

# Reporte de Evaluación RAG para el asistente academico de Desarrollo de Software 1

Métrica	Puntaje Promedio	Tasa de Éxito
Answer Relevancy	0.85	93.2%
Contextual Precision	0.82	86.5%
Contextual Recall	0.86	87.3%
Contextual Relevancy	0.46	47.0%

## Caso de Prueba: 0 - Éxito

### Input del Usuario:

¿Cuáles son los conceptos fundamentales de la ingeniería de software que se deben entender para desarrollar aplicaciones de software efectivas?

### Respuesta Generada:

Según los documentos proporcionados, los conceptos fundamentales de la ingeniería de software que se deben entender para desarrollar aplicaciones de software efectivas son: 1. **Gestión de proyectos**: Aprender a planificar, organizar y controlar el desarrollo de software para entregarlo a tiempo y dentro del presupuesto. 2. **Ciclo de vida del desarrollo de software**: Entender el proceso estructurado que guía el desarrollo del software desde la concepción hasta la implementación y el mantenimiento, incluyendo fases como planificación, análisis, diseño, codificación, pruebas y despliegue. 3. **Aplicación de principios de ingeniería**: Utilizar conceptos de ingeniería como el análisis de requisitos, el diseño, la construcción, las pruebas y el mantenimiento para garantizar que el software sea confiable, eficiente y escalable. 4. **Calidad del software**: Centrarse en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable. 5. **Metodologías y prácticas**: Emplear diversas metodologías (agile, cascada, etc.) y prácticas (control de versiones, integración continua, etc.) para gestionar el desarrollo de software de manera efectiva. 6. **Trabajo en equipo**: Fomentar la colaboración entre equipos multidisciplinarios, incluyendo desarrolladores, analistas, diseñadores, testers y otros profesionales para lograr los objetivos del proyecto. 7. **Análisis de requisitos**: Entender el problema o necesidad de los usuarios y escribir los requisitos de un sistema. 8. **Diseño**: Describir o especificar procesos, componentes e interfaces que representan una solución para el sistema a construir. 9. **Estimación de tiempos y costos**: Entender cómo estimar el tiempo y valor de un producto de software en desarrollo. 10. **Desarrollo de software**: Aprender a aplicar conceptos, métodos, herramientas y prácticas para crear aplicaciones de software efectivas. Espero que esta respuesta sea clara y útil para tu aprendizaje. ¡Si tienes más preguntas, no dudes en preguntar!

### Respuesta Esperada:

Según el contexto, los conceptos fundamentales de la ingeniería de software que se deben entender para desarrollar aplicaciones de software efectivas son: \* Los conceptos fundamentales de la ingeniería de software (no se especifican cuáles son, pero se menciona que es un objetivo entenderlos). \* Los roles y las actividades que se realizan en las etapas de análisis, diseño, codificación, pruebas y despliegue de una aplicación de software. \* Los conceptos principales de las metodologías tradicionales de desarrollo de software.

### **Contexto Relevante:**

1. Objetivos del curso Objetivo general Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### **Contexto Recuperado:**

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

2. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

3. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is highly relevant, but slightly derailed by mentioning the importance of teamwork, which isn't a fundamental concept of software engineering.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the top-ranked node clearly addressing the question, the second node providing key software engineering concepts, and the third node defining software engineering and its focus on high-quality software.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in retrieval context, with all sentences being directly attributed to the 1st node, showcasing a flawless understanding of the software engineering concepts.

Contextual Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the retrieval context has some relevant statements that align with the input, such as 'entender los conceptos fundamentales de la ingeniería de software' and 'conceptos aspectos clave de la ingeniería de software', but also contains irrelevant statements that are not directly related to the fundamental concepts of software engineering, as mentioned in the reasons for irrelevancy.

## Caso de Prueba: 1 - Fallo

### Input del Usuario:

¿Cuáles son los beneficios clave de utilizar metodologías ágiles en el desarrollo de aplicaciones de software?

### Respuesta Generada:

Según el contexto proporcionado, los beneficios clave de utilizar metodologías ágiles en el desarrollo de aplicaciones de software son: 1. **Flexibilidad y Adaptabilidad**: Las metodologías ágiles permiten adaptarse rápidamente a los cambios en los requisitos del proyecto, lo que conduce a una mayor flexibilidad en el desarrollo. 2. **Entregas rápidas y frecuentes**: Las metodologías ágiles se centran en la entrega de pequeñas partes funcionales del software en iteraciones cortas, lo que permite a los usuarios obtener valor del software más rápido. 3. **Mejora de la calidad**: Las metodologías ágiles enfatizan la pruebas y el monitoreo continuo, lo que lleva a una mayor calidad del software. 4. **Colaboración y Comunicación**: Las metodologías ágiles fomentan la colaboración y la comunicación entre el equipo de desarrollo, los stakeholders y los usuarios, lo que conduce a una mayor comprensión de los requisitos y una mayor satisfacción del usuario. 5. **Reducción de Costos**: Las metodologías ágiles permiten estimar y controlar mejor los costos del proyecto, lo que reduce el riesgo de sobrecostos y permite una mayor eficiencia en el uso de los recursos. 6. **Mejora de la Productividad**: Las metodologías ágiles permiten a los equipos de desarrollo trabajar de manera más eficiente, lo que conduce a una mayor productividad y una entrega más rápida del software. 7. **Incremento de la Transparencia**: Las metodologías ágiles permiten una mayor transparencia en el progreso del proyecto, lo que permite a los stakeholders y los usuarios estar informados y tomar decisiones informadas. En resumen, las metodologías ágiles permiten una mayor flexibilidad, entrega rápida y frecuente, mejora de la calidad, colaboración y comunicación, reducción de costos, mejora de la productividad y incremento de la transparencia en el desarrollo de aplicaciones de software.

### Respuesta Esperada:

Según el contexto, los beneficios clave de utilizar metodologías ágiles en el desarrollo de aplicaciones de software son ser un miembro efectivo y eficiente en un equipo de desarrollo ágil.

### Contexto Relevante:

1. Objetivos del curso Objetivo general Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### Contexto Recuperado:

1. estimacion de tiempos y costos en marco de las metodologias agiles costo recursos invertidos en el desarrollo de la aplicacion precio cantidad de dinero que un usuario paga para acceder a la aplicacio.....

2. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....

3. modelo de despliegue como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones ram procesadores espacio de almacenamiento ancho de banda herramientas y monitoreo monitoreo.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all the top-ranked nodes in the retrieval contexts, such as the 1st node about 'estimacion de tiempos y costos', the 2nd node about 'estimacion de tiempos y costos', and the 3rd node about 'modelo de despliegue', are irrelevant to the input, and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.18 (Threshold: 0.5) - The score is 0.18 because the retrieval context mostly discusses unrelated topics such as estimating time and costs, resource costs, and server management, which are not directly related to the benefits of using agile methodologies, as stated in the input.

## Caso de Prueba: 2 - Éxito

### Input del Usuario:

¿Cuál es el objetivo principal de la ingeniería de software en el desarrollo de aplicaciones?

### Respuesta Generada:

Según el Documento 6, el objetivo principal de la ingeniería de software es la "creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios".

### Respuesta Esperada:

El objetivo principal de la ingeniería de software en el desarrollo de aplicaciones es proporcionar las bases conceptuales fundamentales y los elementos metodológicos necesarios para llevar a cabo el desarrollo de aplicaciones de software.

### Contexto Relevante:

1. Objetivos del curso Objetivo general Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....

### Contexto Recuperado:

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....

2. presentación de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escritas.....

3. objetivos del curso objetivos específicos entender cómo estimar un producto de software tiempo y valor en una aplicación de software desarrollar una aplicación de software aplicar algunos modelos.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question about the main objective of software engineering in application development!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which only provides general information about the course, and node 3 which lists specific objectives of a course, are correctly ranked lower than the relevant node 1 which clearly states the main objective of software engineering in application development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which provides a matching conceptual

foundation for software engineering in application development.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context has many irrelevant statements, such as 'remembering ideas from today's class' and 'identifying good practices in web application development', which are not related to the main objective of software engineering in application development, and only a few relevant statements, such as 'providing the student with the fundamental conceptual bases of software engineering' and 'applying concepts, methods, and tools to create software applications'.

## Caso de Prueba: 3 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal de la etapa de análisis en el desarrollo de una aplicación de software?

### Respuesta Generada:

Basándome en el contexto proporcionado, específicamente en el Documento 9, que habla sobre los conceptos clave de la ingeniería de software, puedo responder que el objetivo principal de la etapa de análisis en el desarrollo de una aplicación de software es entender los requisitos del sistema y definir adecuadamente los requerimientos del proyecto. En otras palabras, la etapa de análisis se centra en recopilar, analizar y documentar los requisitos del sistema, para luego definir los límites y los objetivos del proyecto. Esto permitirá a los desarrolladores crear un plan de trabajo efectivo y establecer prioridades para el desarrollo de la aplicación. Además, como se menciona en el Documento 2, entender claramente el objetivo del proyecto y definir adecuadamente los requerimientos es crucial para realizar una estimación de costos y tiempo de desarrollo precisa y crear un plan de trabajo efectivo.

### Respuesta Esperada:

No se menciona explícitamente el objetivo principal de la etapa de análisis en el desarrollo de una aplicación de software en el contexto proporcionado. Sin embargo, se puede inferir que el objetivo principal de la etapa de análisis es entender y aplicar las actividades que se realizan en esa etapa, ya que se menciona como uno de los objetivos específicos del curso.

### Contexto Relevante:

1. Objetivos del curso Objetivo general Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### Contexto Recuperado:

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

2. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....

3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 mentioning a general project objective and node 3 discussing agile development principles, are correctly ranked lower than the relevant node 1, which directly addresses the main objective of the analysis stage.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be accurately attributed to the 1st node in the retrieval context, which clearly outlines the objective of the analysis stage.

Contextual Relevancy: 0.19 (Threshold: 0.5) - The score is 0.19 because the retrieval context is mostly irrelevant, with statements like 'realizar la estimacion de costos y tiempo de desarrollo' and 'establecer prioridades' not directly related to the main objective of the analysis stage in software development, but a few relevant statements like 'entender y aplicar los roles y las actividades que se realizan en las etapas de analisis diseno codificacion pruebas y despliegue de una aplicacion de software' and 'definir adecuadamente los requerimientos' were found.



## Caso de Prueba: 4 - Fallo

### Input del Usuario:

¿Cuáles son los conceptos principales de las metodologías tradicionales de desarrollo de software?

### Respuesta Generada:

Según el documento 4, los conceptos principales de las metodologías tradicionales de desarrollo de software incluyen RUP (Rational Unified Process) y MSF (Microsoft Solutions Framework). Estas metodologías se enfocan en un enfoque más estructurado y predicable para el desarrollo de software, a diferencia de las metodologías ágiles que se enfocan en la flexibilidad y la adaptabilidad. Es importante destacar que, según el documento 1 y 3, el curso de Desarrollo de Software tiene como objetivo entender los conceptos fundamentales de la ingeniería de software, incluyendo las metodologías tradicionales de desarrollo de software. Sin embargo, no se proporciona más información específica sobre los conceptos principales de estas metodologías en los documentos proporcionados.

### Respuesta Esperada:

Los conceptos principales de las metodologías tradicionales de desarrollo de software no se especifican explícitamente en el contexto proporcionado. Sin embargo, se menciona que es uno de los objetivos específicos del curso, lo que implica que se abordarán en el curso, pero no se proporciona información adicional sobre qué son esos conceptos.

### Contexto Relevante:

1. Objetivos del curso Objetivo general Proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### Contexto Recuperado:

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

2. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....

3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output partially addresses the main concepts of traditional software development methodologies, but is distracted by agile

methodologies and course objectives, which are not the focus of the input.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions understanding traditional software development methodologies as a course objective, is correctly ranked higher than the second node, which only lists software development aspects without explicit information about main concepts, and the third node, which is a duplicate of the first node, is also correctly ranked high.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the output mentions 'conceptos principales de las metodologías tradicionales de desarrollo de software', which is related to nodes 1 and 3 in the retrieval context, but lacks specific details that can be attributed to the context.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context is mostly irrelevant, with many statements focusing on agile development, software engineering concepts, and course objectives, as stated in the reasons for irrelevancy, such as 'The input asks about traditional methodologies, but this statement is about agile practices.' and 'The statement is about the course objectives, not about traditional software development methodologies.'

## Caso de Prueba: 5 - Fallo

### Input del Usuario:

¿Cuál es el modelo de estimación de software más adecuado para un proyecto de desarrollo de una aplicación web que requiere un tiempo de entrega corto y un presupuesto limitado?

### Respuesta Generada:

Basándome en los documentos proporcionados, puedo concluir que el modelo de estimación de software más adecuado para un proyecto de desarrollo de una aplicación web que requiere un tiempo de entrega corto y un presupuesto limitado es el desarrollo de software ágil. El desarrollo ágil se centra en la entrega incremental y continua de software con valor, lo que permite adaptarse rápidamente a los cambios en los requisitos del proyecto (Documento 9 y 10). Además, este enfoque iterativo permite entregar software funcional en períodos de tiempo cortos, lo que se ajusta a las necesidades del proyecto de entrega corta (Documento 9). El desarrollo ágil también enfatiza la colaboración en equipo, la planificación continua y el aprendizaje continuo, lo que facilita la gestión del proyecto y la toma de decisiones informadas (Documento 6). Esto es especialmente importante en proyectos con presupuesto limitado, ya que se pueden priorizar las características más valiosas y enfocar los recursos en las actividades más críticas. Además, el desarrollo ágil permite una mayor flexibilidad y adaptabilidad en el proyecto, lo que es beneficioso cuando se enfrentan cambios en los requisitos o limitaciones en el presupuesto (Documento 10). En resumen, el desarrollo ágil es el modelo de estimación de software más adecuado para un proyecto de desarrollo de una aplicación web que requiere un tiempo de entrega corto y un presupuesto limitado, ya que se centra en la entrega incremental de software con valor, la colaboración en equipo y la adaptabilidad al cambio.

### Respuesta Esperada:

Basándome en el contexto proporcionado, no hay información específica sobre modelos de estimación de software. Sin embargo, considerando que el proyecto requiere un tiempo de entrega corto y un presupuesto limitado, puedo inferir que se necesita un enfoque ágil y flexible. En este sentido, un modelo de estimación de software adecuado podría ser el enfoque de "Story Points" o "Puntos de Historia", que se utiliza en el marco de trabajo Ágil. Este enfoque se centra en estimar la complejidad relativa de las tareas en lugar de intentar proporcionar una estimación precisa de tiempo o costo. Esto permite a los equipos de desarrollo adaptarse rápidamente a los cambios y priorizar las tareas según sea necesario. Sin embargo, es importante tener en cuenta que este contexto no proporciona información suficiente para dar una respuesta definitiva, y se requiere más información sobre el proyecto y sus necesidades específicas para elegir el modelo de estimación más adecuado.

### Contexto Relevante:

1. Objetivos del curso  
Objetivos específicos  
Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)  
Desarrollar una aplicación de software.  
Aplicar algunos mo.....

### Contexto Recuperado:

1. objetivos del curso  
objetivos especificos  
entender como estimar un producto de software  
tiempo y valor en de una aplicacion de software  
desarrollar una aplicacion de software  
aplicar algunos modelos

.....

2. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos

.....

3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer without any irrelevant statements.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, and it's ranked third. The first node, ranked 1, is irrelevant because it talks about course objectives, and the second node, ranked 2, is also irrelevant as it only discusses project objectives and planning, whereas the relevant node, ranked 3, provides information about agile development related to the 'Story Points' estimation model.

Contextual Recall: 0.33 (Threshold: 0.5) - The score is 0.33 because the response only partially relates to the node 3 in the retrieval context, mentioning 'enfoque ágil', but lacks specific connections to the project's needs and requirements, leading to a lower score.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context mostly contains irrelevant statements that are too broad or focus on general software development, agile methodology, or good practices, as stated in the reasons for irrelevancy, whereas only a few statements, such as 'realizar la estimacion de costos y tiempo de desarrollo', are actually relevant to the input's specific question about software estimation models for web application development.

## Caso de Prueba: 6 - Fallo

### Input del Usuario:

¿Cuál es la importancia de estimar el tiempo y el valor en dólares de una aplicación de software en el desarrollo de proyectos de aplicación de software?

### Respuesta Generada:

La importancia de estimar el tiempo y el valor en dólares de una aplicación de software en el desarrollo de proyectos de aplicación de software radica en varios aspectos: 1. **Planificación y coordinación**: La estimación del tiempo y el valor monetario de una aplicación de software permite planificar y coordinar adecuadamente el desarrollo del proyecto, asignando recursos y estableciendo fechas límite realistas. 2. **Gestión de recursos**: La estimación de costos directos e indirectos permite gestionar adecuadamente los recursos monetarios y humanos necesarios para el desarrollo del proyecto, evitando gastos innecesarios y optimizando la inversión. 3. **Valor para el usuario**: La estimación del valor en dinero que un usuario pagaría por utilizar la aplicación de software permite evaluar la viabilidad del proyecto y determinar si el desarrollo es rentable. 4. **Beneficio económico neto**: La estimación del valor en dinero permite evaluar el beneficio económico neto que se obtendrá del desarrollo y comercialización de la aplicación de software. 5. **Tomar decisiones informadas**: La estimación del tiempo y el valor en dólares permite tomar decisiones informadas sobre el desarrollo del proyecto, como si es rentable continuar o no con el proyecto, y si es necesario realizar ajustes en el presupuesto o en el alcance del proyecto. 6. **Identificar riesgos**: La estimación del tiempo y el valor en dólares permite identificar posibles riesgos y desafíos en el desarrollo del proyecto, lo que permite tomar medidas para mitigarlos. 7. **Mejora de la comunicación**: La estimación del tiempo y el valor en dólares facilita la comunicación entre stakeholders, como desarrolladores, product owners, y clientes, sobre los objetivos y los recursos necesarios para el desarrollo del proyecto. En resumen, la estimación del tiempo y el valor en dólares de una aplicación de software es crucial para planificar, gestionar y desarrollar proyectos de aplicación de software de manera efectiva y rentable.

### Respuesta Esperada:

La importancia de estimar el tiempo y el valor en dólares de una aplicación de software en el desarrollo de proyectos de aplicación de software es que permite entender cómo estimar un producto de software, lo que es un objetivo específico del curso.

### Contexto Relevante:

1. Objetivos del curso  
Objetivos específicos  
Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)  
Desarrollar una aplicación de software.  
Aplicar algunos mo.....

### Contexto Recuperado:

1. objetivos del curso  
objetivos especificos  
entender como estimar un producto de software  
tiempo y valor en de una aplicacion de software  
desarrollar una aplicacion de software  
aplicar algunos modelos  
.....

2. estimacion de tiempos y costos en marco de las metodologias agiles  
costo recursos invertidos en el desarrollo de la aplicacion  
precio cantidad de dinero que un usuario paga para acceder a la

aplicacio.....

3. sprint planning en esta reunion participan de manera colaborativa todo el equipo scrum scrum master product owner y equipo de desarrollo practicas agiles en scrum el tiempo de esta reunion es de maxim.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 which clearly addresses the question, and node 2 which explains the importance of 'estimacion de tiempos y costos', are ranked higher than the irrelevant node 3, which is about 'sprint planning' and does not provide any information about estimating time and value in software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless connection between the output and the retrieval context.

Contextual Relevancy: 0.47 (Threshold: 0.5) - The score is 0.47 because the retrieval context is mostly irrelevant, with statements like 'The statement is talking about applying models, tools, languages, and platforms to implement a software application...' and 'The statement talks about the role of the scrum master...' not related to estimating time and value in dollars of a software application, but a few relevant statements like 'entender como estimar un producto de software tiempo y valor en de una aplicacion de software' and 'estimacion de tiempos y costos en marco de las metodologias agiles' bring the score up.

## Caso de Prueba: 7 - Éxito

### Input del Usuario:

¿Cuál es el modelo de desarrollo de software que se centra en la entrega continua y flexible de pequeñas partes del producto, permitiendo una retroalimentación temprana y una respuesta rápida a los cambios en los requisitos?

### Respuesta Generada:

Según los documentos proporcionados, el modelo de desarrollo de software que se centra en la entrega continua y flexible de pequeñas partes del producto, permitiendo una retroalimentación temprana y una respuesta rápida a los cambios en los requisitos es el desarrollo de software ágil. En particular, los documentos 1, 2, 3 y 4 mencionan que el desarrollo ágil de software se basa en procesos ágiles, que enfatizan la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo. Esto permite una entrega temprana y continua de software con valor, satisfaciendo al cliente y permitiendo la retroalimentación y el ajuste a los cambios en los requisitos. Además, el documento 4 menciona los 12 principios del Manifiesto Ágil, que incluyen la entrega temprana y continua de software con valor, la colaboración entre responsables de negocio y desarrolladores, y la confianza en los individuos motivados para ejecutar el trabajo. En resumen, el desarrollo de software ágil es el modelo que se centra en la entrega continua y flexible de pequeñas partes del producto, permitiendo una retroalimentación temprana y una respuesta rápida a los cambios en los requisitos.

### Respuesta Esperada:

No se menciona explícitamente el modelo de desarrollo de software que se centra en la entrega continua y flexible de pequeñas partes del producto en el contexto proporcionado. Sin embargo, se puede inferir que se refiere a Agile, ya que se habla de "buena prácticas aplicables a cada momento de un proceso de desarrollo de aplicaciones web" y Agile se centra en la entrega continua y flexible de pequeñas partes del producto, permitiendo una retroalimentación temprana y una respuesta rápida a los cambios en los requisitos.

### Contexto Relevante:

1. Objetivos del curso  
Objetivos específicos  
Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)  
Desarrollar una aplicación de software.  
Aplicar algunos mo.....

### Contexto Recuperado:

1. desarrollo de software ágil  
el desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles se concibieron como una alternativa a las pract.....
2. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.71 (Threshold: 0.5) - The score is 0.71 because the response is mostly relevant, but gets sidetracked by mentioning the 12 principles of the Agile Manifesto, which, although related, are not directly addressing the model of software development asked about.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node mentioning Agile development, the second node explicitly mentioning incremental delivery, and the third node again highlighting incremental delivery, making all nodes perfectly ranked.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be confidently attributed to nodes 1 and 2 in the retrieval context, which explicitly mention Agile development and its core principles.

Contextual Relevancy: 0.91 (Threshold: 0.5) - The score is 0.91 because the retrieval context, although containing irrelevant information about a specific organization, provided highly relevant statements about agile software development, such as 'agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental', which directly relate to the input.



## Caso de Prueba: 8 - Fallo

### Input del Usuario:

¿Cuáles son las herramientas y modelos más comunes utilizados para estimar el tiempo y costo de desarrollo de una aplicación de software?

### Respuesta Generada:

Según los documentos proporcionados, los modelos y herramientas más comunes utilizados para estimar el tiempo y costo de desarrollo de una aplicación de software en el marco de las metodologías ágiles son: **\*\*Modelos de estimación:\*\*** 1. Estimación basada en analogía (Documento 4): se utiliza para calcular el costo estimado de un proyecto mediante la velocidad del equipo de desarrollo y el costo de operación mensual. 2. Planning Poker (Documento 10): técnica colaborativa para estimar historias de usuario asignando puntos a cada tarea. **\*\*Herramientas de estimación:\*\*** 1. Jira (Documento 7): herramienta de gestión para estimar y tracking el progreso del proyecto. 2. Taiga (Documento 7): herramienta de gestión para estimar y tracking el progreso del proyecto. 3. GitHub (Documento 7): plataforma de control de versiones para estimar y tracking el progreso del proyecto. **\*\*Costos directos e indirectos:\*\*** 1. Costos directos (Documento 1): aquellos que se pueden atribuir directamente al proyecto de software, como salarios y beneficios del personal, contratos de consultores, herramientas y software para el desarrollo, infraestructura y equipos, capacitación y formación, gastos de viaje, pruebas y control de calidad, despliegue e implementación en infraestructuras. 2. Costos indirectos (Documento 1): aquellos que no se pueden atribuir directamente a un proyecto específico pero son necesarios para el funcionamiento general y el soporte de la empresa que desarrolla el software, como administración general y gestión personal administrativo, infraestructura y servicios generales, agua, luz, teléfono, marketing y ventas, seguros, impuestos, etc. En resumen, los modelos y herramientas más comunes utilizados para estimar el tiempo y costo de desarrollo de una aplicación de software son la estimación basada en analogía, Planning Poker, Jira, Taiga, GitHub, y la consideración de costos directos e indirectos.

### Respuesta Esperada:

Según el contexto, no se proporciona una lista específica de herramientas y modelos comunes utilizados para estimar el tiempo y costo de desarrollo de una aplicación de software. Sin embargo, se menciona que se aprenderán "algunos modelos, herramientas, lenguajes y plataformas" para implementar una aplicación de software.

### Contexto Relevante:

1. Objetivos del curso  
Objetivos específicos  
Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)  
Desarrollar una aplicación de software. Aplicar algunos mo.....

### Contexto Recuperado:

1. estimacion de tiempos y costos en marco de las metodologias agiles  
estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables  
costos directos son aq.....

2. modelo de despliegue como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones ram procesadores espacio de almacenamiento ancho de banda herramientas y monitoreo monitoreo.....

3. objetivos del curso objetivos específicos entender como estimar un producto de software tiempo y valor en de una aplicación de software desarrollar una aplicación de software aplicar algunos modelos .....

### **Métricas de Evaluación:**

Answer Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the output provided some relevant information, but was also sidetracked by unnecessary explanations about GitHub and the financial aspect of software development, which didn't directly address the input question.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because the first two nodes in the retrieval contexts, ranked 1 and 2, are irrelevant as they don't provide a specific list of tools and models for estimating time and cost of software development, and instead talk about deployment and server resources, respectively, which should be ranked lower than the third node, ranked 3, which mentions learning some models, tools, languages, and platforms for implementing a software application, which is related to the question.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the first sentence is partially attributed to the 3rd node in the retrieval context, which mentions 'aplicar algunos modelos, herramientas, lenguajes y plataformas para implementar una aplicación de software...', but the second sentence lacks any matching information in the retrieval context.

Contextual Relevancy: 0.69 (Threshold: 0.5) - The score is 0.69 because the retrieval context is only partially relevant to the input, as it contains some relevant statements about estimating time and cost of software development, such as 'estimación de tiempos y costos en marco de las metodologías ágiles' and 'entender como estimar un producto de software tiempo y valor', but also includes many irrelevant statements about deployment, good practices, and other unrelated topics.

## Caso de Prueba: 9 - Éxito

### Input del Usuario:

¿Cuáles son las buenas prácticas para estimar el tiempo y valor en \$\$\$ de una aplicación de software?

### Respuesta Generada:

Según los documentos proporcionados, las buenas prácticas para estimar el tiempo y valor en \$\$\$ de una aplicación de software en el marco de las metodologías ágiles son: 1. **\*\*Estimación basada en datos disponibles, experiencia pasada y suposiciones razonables\*\*** (Documento 3): La estimación debe ser informada y basada en datos concretos, experiencia previa y suposiciones razonables. 2. **\*\*Identificar costos directos e indirectos\*\*** (Documento 3): Es importante identificar los costos directos, como salarios, contratos de consultores, herramientas y software para el desarrollo, infraestructura, capacitación, gastos de viaje, pruebas y control de calidad, despliegue e implementación. También es fundamental considerar los costos indirectos, como administración general, infraestructura y servicios generales, marketing y ventas, seguros, impuestos, etc. 3. **\*\*Utilizar técnicas de estimación ágil\*\*** (Documento 7): La estimación ágil se basa en la estimación relativa y utiliza técnicas colaborativas, como Planning Poker (Documento 10), para asignar puntos a las tareas o historias de usuario. Esto permite cuantificar la magnitud del proyecto y las tareas que se deben asumir, y obtener una idea aproximada del tiempo y recursos que se van a consumir. 4. **\*\*Considerar el valor del proyecto\*\*** (Documento 2): El valor se refiere a los beneficios que los usuarios obtienen al utilizar la aplicación. La estimación del valor debe considerar la utilidad, el beneficio económico neto que se obtiene del desarrollo y comercialización de la aplicación. 5. **\*\*Definir y dividir el proyecto en iteraciones\*\*** (Documento 4): La metodología ágil Scrum divide el proyecto en iteraciones llamadas sprints, lo que permite trabajar colaborativamente en equipo y obtener el mejor resultado posible. Es importante destacar que las estimaciones son solo valores aproximados y no valores exactos. Es fundamental considerar las limitaciones y la incertidumbre inherente en la estimación de tiempos y costos en el desarrollo de software.

### Respuesta Esperada:

Según el contexto, las buenas prácticas para estimar el tiempo y valor en \$\$\$ de una aplicación de software no se especifican explícitamente. Sin embargo, se menciona que uno de los objetivos del curso es "Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)". Esto sugiere que se abordarán buenas prácticas para estimar el tiempo y valor en \$\$\$ de una aplicación de software en el curso, pero no se proporcionan detalles específicos.

### Contexto Relevante:

1. Objetivos del curso  
Objetivos específicos  
Entender como estimar un producto de software (tiempo y valor en \$\$\$ de una aplicación de software)  
Desarrollar una aplicación de software.  
Aplicar algunos mo.....

### Contexto Recuperado:

1. objetivos del curso  
objetivos especificos  
entender como estimar un producto de software  
tiempo y valor en de una aplicacion de software  
desarrollar una aplicacion de software  
aplicar algunos modelos  
.....

2. estimacion de tiempos y costos en marco de las metodologias agiles costo recursos invertidos en el desarrollo de la aplicacion precio cantidad de dinero que un usuario paga para acceder a la aplicacio.....

3. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.89 (Threshold: 0.5) - The score is 0.89 because the response is mostly relevant, but slightly derailed by an unnecessary mention of Scrum methodology, which is not directly related to estimating time and value of a software application.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are perfectly relevant to the input, providing direct explanations and concepts related to estimating time and value in \$\$\$ of a software application.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes in retrieval context, specifically nodes 1 and 2, which explicitly mention estimating time and value in \$\$\$ of a software application.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context provides relevant statements about estimating time and value in software application development, such as 'entender como estimar un producto de software tiempo y valor en de una aplicacion de software' and 'estimacion de tiempos y costos en marco de las metodologias agiles', despite some irrelevant statements about developing a software application.

## Caso de Prueba: 10 - Éxito

### Input del Usuario:

¿Cuál es la principal diferencia entre un modelo de ciclo de vida de desarrollo de software tradicional y uno ágil?

### Respuesta Generada:

La principal diferencia entre un modelo de ciclo de vida de desarrollo de software tradicional y uno ágil radica en la manera en que se aborda el desarrollo del software. En un modelo de ciclo de vida tradicional, el desarrollo de software se aborda de manera secuencial y Waterfall (Cascada), es decir, cada fase se completa antes de pasar a la siguiente. Esto significa que se realizan planes detallados y se establecen fechas de entrega fijas al principio del proyecto. El enfoque tradicional se centra en la planificación, el análisis y el diseño detallados antes de comenzar a codificar. Por otro lado, en un modelo de ciclo de vida ágil, el desarrollo de software se aborda de manera iterativa e incremental. Se dividen las tareas en pequeñas iteraciones o Sprints, y se entrega el software funcionando al final de cada iteración. El enfoque ágil se centra en la entrega rápida y continua de software funcionando, y se adapta a los cambios y nuevos requisitos que surjan durante el desarrollo. En resumen, la principal diferencia es que el enfoque tradicional se centra en la planificación y el diseño detallados antes de comenzar a codificar, mientras que el enfoque ágil se centra en la entrega rápida y continua de software funcionando, con adaptación a los cambios y nuevos requisitos. Este concepto se encuentra en los Documentos 2, 4, 7, 9 y 10.

### Respuesta Esperada:

La principal diferencia entre un modelo de ciclo de vida de desarrollo de software tradicional y uno ágil es la forma en que se abordan los requerimientos y la planificación del proyecto. En un modelo tradicional, se define y se planifica todo el proyecto al principio, mientras que en un modelo ágil, se trabajan en pequeñas iteraciones, con una planificación y priorización continua de los requerimientos, a través de prácticas como el Product Backlog y el Sprint.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarroll.....

### Contexto Recuperado:

1. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stakeholders.....

2. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologías de desarrollo de .....

3. presentación de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias

escri.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.91 (Threshold: 0.5) - The score is 0.91 because the response is highly relevant to the input question, but slightly derailed by an unnecessary reference to external documents.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which explains the concept of software development life cycle, and node 2 which mentions agile methodologies and practices, are ranked higher than the irrelevant node 3 which only introduces the concept of software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which discusses traditional and agile software development models and methodologies.

Contextual Relevancy: 0.65 (Threshold: 0.5) - The score is 0.65 because the retrieval context contains many irrelevant statements, such as 'lluvia de ideas', 'ideacion concepto ciclo de vida del software hoy dia', and 'presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software', which are not related to the main difference between traditional and agile software development life cycles, but it also contains some relevant statements like 'ciclo de vida del desarrollo de software', 'metodologias agiles para el desarrollo de software', and 'metodologias tradicionales' that support the input.

## Caso de Prueba: 11 - Éxito

### Input del Usuario:

¿Cuáles son los modelos de ciclo de vida del software más comunes y cómo se comparan entre sí?

### Respuesta Generada:

Según los documentos proporcionados, los modelos de ciclo de vida del software más comunes son:

1. **Modelo Cascada**: Este modelo sigue una secuencia lineal de fases, desde la concepción hasta la implementación y el mantenimiento. Cada fase se completa antes de pasar a la siguiente. Es un modelo tradicional y rígido.
2. **Modelo Ágil**: Este modelo se basa en la entrega incremental y iterativa del software, con una gran colaboración entre el equipo de desarrollo y los stakeholders. Se enfatiza en la flexibilidad y la capacidad de respuesta a los cambios.
3. **Scrum**: Es una metodología ágil que se centra en la entrega en sprint (períodos cortos de trabajo) y la revisión continua del progreso. Tiene roles definidos como Product Owner, Scrum Master y Development Team.
4. **XP (Extreme Programming)**: Es una metodología ágil que se centra en la programación en parejas, la refactorización del código y la entrega continua. Se enfatiza en la calidad del código y la colaboración del equipo.
5. **Kanban**: Es una metodología ágil que se centra en la visualización del flujo de trabajo, la limitación del trabajo en proceso y la entrega continua. No tiene roles definidos ni sprint.

Comparación entre sí:

- \* **Rigidez**: El modelo Cascada es más rígido y lineal, mientras que los modelos Ágil, Scrum, XP y Kanban son más flexibles y adaptables a los cambios.
- \* **Tiempo de entrega**: Los modelos Ágil, Scrum, XP y Kanban se centran en la entrega incremental y continua, mientras que el modelo Cascada se enfoca en la entrega final al final del proyecto.
- \* **Colaboración**: Los modelos Ágil, Scrum, XP y Kanban enfatizan la colaboración entre el equipo de desarrollo y los stakeholders, mientras que el modelo Cascada tiene una estructura más jerárquica.
- \* **Flexibilidad**: Los modelos Ágil, Scrum, XP y Kanban son más flexibles y permiten cambios en el proyecto, mientras que el modelo Cascada es más rígido y dificulta los cambios.

En resumen, cada modelo de ciclo de vida del software tiene sus ventajas y desventajas, y se deben elegir según las necesidades y requerimientos del proyecto.

### Respuesta Esperada:

Los modelos de ciclo de vida del software más comunes son:

1. **Metodologías tradicionales**: se caracterizan por seguir un enfoque secuencial y predeterminado, donde cada fase se completa antes de pasar a la siguiente. Algunos ejemplos de artefactos de estas metodologías son los diagramas de flujo, los diagramas de Gantt y los informes de avance.
2. **Metodologías Ágiles**: se centran en la flexibilidad y la adaptabilidad, permitiendo cambios en el proyecto a medida que avanza. Algunos ejemplos de metodologías ágiles son Scrum, XP y Kanban. Estas metodologías se basan en la iteración y la entrega continua, y utilizan herramientas como el Product Backlog para gestionar los requisitos del proyecto. En comparación, las metodologías tradicionales son más rígidas y secuenciales, mientras que las metodologías ágiles son más flexibles y adaptables. Las metodologías ágiles se centran en la entrega continua y la colaboración del equipo, mientras que las metodologías tradicionales se centran en la planificación y el control.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarroll.....

### **Contexto Recuperado:**

1. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologías de desarrollo de .....
2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stake.....
3. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a comprehensive and accurate comparison of the most common software development life cycle models!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are highly relevant to the input question, providing comprehensive overviews and explanations of software development life cycle models, stages, and methodologies.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the nodes in retrieval context, with the 1st node mentioning key terms like 'modelos de ciclo de vida del software', 'metodologías de desarrollo de software', 'metodologías ágiles', and 'metodologías tradicionales' that match the sentences in the expected output.

Contextual Relevancy: 0.79 (Threshold: 0.5) - The score is 0.79 because the retrieval context contains some relevant information about software development life cycle models, such as 'ciclo de vida del desarrollo de software', 'metodologías ágiles para el desarrollo de software', and 'aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software', but is largely irrelevant due to the presence of unrelated information about 'world wide web', 'desarrollo de software i', and 'calidad del software'.



## Caso de Prueba: 12 - Éxito

### Input del Usuario:

¿Cuál es la relación entre el ciclo de vida del software y las metodologías de desarrollo?

### Respuesta Generada:

De acuerdo con los documentos proporcionados, la relación entre el ciclo de vida del software y las metodologías de desarrollo es que las metodologías de desarrollo de software establecen un marco estructurado para guiar el proceso de desarrollo de software, que se encuentra dentro del ciclo de vida del software. El ciclo de vida del software se refiere a las etapas que describen el proceso de desarrollo de software, desde su concepción hasta su puesta en producción. Estas etapas incluyen la planificación, el análisis, el diseño, la codificación, las pruebas y el despliegue. Por otro lado, las metodologías de desarrollo de software son conjuntos estructurados de prácticas, principios y procedimientos que guían y regulan el proceso de creación, implementación, despliegue y mantenimiento de sistemas de software. Estas metodologías establecen un marco para organizar y gestionar las tareas involucradas en cada fase del ciclo de vida del software. En otras palabras, las metodologías de desarrollo de software definen cómo se llevarán a cabo las etapas del ciclo de vida del software. Por ejemplo, una metodología ágil como Scrum establece un marco para organizar y gestionar las tareas involucradas en la planificación, el análisis, el diseño, la codificación, las pruebas y el despliegue del software. En resumen, el ciclo de vida del software es el proceso general de desarrollo de software, mientras que las metodologías de desarrollo de software son los enfoques específicos para llevar a cabo ese proceso de manera estructurada y organizada.

### Respuesta Esperada:

La relación entre el ciclo de vida del software y las metodologías de desarrollo es que las metodologías de desarrollo, como las ágiles (Scrum, XP, Kanban) y las tradicionales, se aplican durante el ciclo de vida del desarrollo del software. El ciclo de vida del software se refiere al proceso de desarrollo de un producto de software, que incluye análisis y especificación de requerimientos, diseño, planeación, desarrollo, pruebas y mantenimiento. Las metodologías de desarrollo brindan un enfoque y un marco para llevar a cabo cada una de estas etapas del ciclo de vida del software de manera efectiva y eficiente.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarrollo.....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stake.....

3. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are highly relevant to the input, with the first node explaining the software development life cycle, the second node detailing the interaction between the life cycle and development methodologies, and the third node clarifying the concept of software development methodology and its relation to the life cycle.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes 2 and 3 in the retrieval context, providing a comprehensive understanding of the software development lifecycle and methodologies.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context has some relevant statements about the software life cycle and development methodologies, but is largely irrelevant due to statements about course presentation, software definitions, and unrelated concepts, as stated in the reasons for irrelevancy, such as 'The statement is talking about the presentation and characteristics of the course...' and 'The statement is defining what software and software engineering are...'.

## Caso de Prueba: 13 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal del análisis de requerimientos en el desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal del análisis de requerimientos en el desarrollo de software es identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software (Documento 2 y Documento 9). En otras palabras, el análisis de requerimientos busca determinar los requisitos del sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave, con el fin de especificar qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. De esta manera, el análisis de requerimientos permite establecer una base sólida para el desarrollo del software, asegurando que se cumplan los objetivos del proyecto y se satisfacen las necesidades del cliente.

### Respuesta Esperada:

El propósito principal del análisis de requerimientos en el desarrollo de software es especificar los requisitos funcionales y no funcionales del producto de software mediante el Product Backlog.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarroll.....

### Contexto Recuperado:

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant response to the question about the main purpose of requirement analysis in software development!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which provide context about understanding user needs and specifying requirements in software development, are ranked higher than the irrelevant node 3, which is about agile development and does not directly address the purpose of requirement analysis in software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly aligns with the 2nd node in the retrieval context, which explicitly discusses the analysis of requirements in software development.

Contextual Relevancy: 0.31 (Threshold: 0.5) - The score is 0.31 because the retrieval context mostly contains irrelevant statements that do not directly relate to the purpose of requirement analysis, as stated in the reasons for irrelevancy, such as 'The statement is talking about design, which is not directly related to the purpose of requirement analysis.' However, there are a few relevant statements that provide information about the purpose of requirement analysis, like 'conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema'.

## Caso de Prueba: 14 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal del Sprint cero en el desarrollo de un producto de software utilizando prácticas ágiles?

### Respuesta Generada:

El propósito principal del Sprint cero en el desarrollo de un producto de software utilizando prácticas ágiles es realizar los preparativos previos a comenzar el desarrollo. Durante el Sprint cero se realizan diferentes tareas, como definir la visión del producto, establecer las necesidades del producto, conformar el equipo de trabajo, definir fechas críticas, crear vistas de diseño del sistema, seleccionar la tecnología a utilizar, crear un release plan, priorizar y estimar los requisitos, y definir los criterios de aceptación. En otras palabras, el Sprint cero es una iteración inicial que permite preparar el terreno para el desarrollo del producto de software, establecer las bases para el resto del proyecto y asegurarse de que todos los involucrados estén alineados y comprometidos con el objetivo del proyecto.

### Respuesta Esperada:

El propósito principal del Sprint cero en el desarrollo de un producto de software utilizando prácticas ágiles es definir la idea, visión del producto, necesidades del producto, equipo de trabajo, fechas críticas, vistas de diseño del sistema, tecnología a usar, release plan, priorización, estimación y criterios de aceptación.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarrollo.....

### Contexto Recuperado:

1. iniciando un proyecto de software usando practicas agiles el sprint cero iteracion cero en algunos equipos es frecuente el uso del llamado sprint cero cuyo objetivo son los preparativos previos a come.....

2. iniciando un proyecto de software usando practicas agiles el sprint cero iteracion cero en algunos equipos es frecuente el uso del llamado sprint cero cuyo objetivo son los preparativos previos a come.....

3. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant answer to the question about the primary purpose of Sprint zero in agile software development!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, directly address the main purpose of Sprint zero, providing a clear and comprehensive understanding of its role in agile software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which explicitly mentions all the necessary software development aspects.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context is mostly irrelevant to the input, with reasons such as containing information about web development, requirement analysis, software architecture, traditional methodologies, and other agile methodologies, which are not related to the purpose of Sprint zero in agile software development, but has some relevant statements that mention Sprint zero and its objectives.

## Caso de Prueba: 15 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal del protocolo HTTP en la comunicación entre clientes y servidores web?

### Respuesta Generada:

Según el contexto proporcionado, especialmente en el Documento 3, el propósito principal del protocolo HTTP en la comunicación entre clientes y servidores web es permitir la transferencia de datos entre el cliente (generalmente un navegador) y el servidor. HTTP es un protocolo sin estado, lo que significa que no guarda ninguna información sobre conexiones anteriores; al finalizar la transacción, todos los datos se pierden. En otras palabras, el propósito principal de HTTP es permitir que los clientes web (como navegadores) soliciten y reciban datos de los servidores web, y que los servidores web respondan a estas solicitudes. Esto se logra mediante métodos como GET y POST, que permiten la transferencia de datos entre el cliente y el servidor.

### Respuesta Esperada:

El propósito principal del protocolo HTTP en la comunicación entre clientes y servidores web es permitir la transferencia de datos entre ellos, específicamente para que los clientes (navegadores web) puedan realizar peticiones a los servidores web y recibir respuestas en forma de páginas web, imágenes, archivos, etc.

### Contexto Relevante:

1. Aspectos a tratar Conceptos Generales. El proceso de desarrollo de software: Ciclo de vida de desarrollo del software, modelos de ciclo de vida del software, conceptos sobre metodologías de desarroll.....

### Contexto Recuperado:

1. servidores y clientes web http y https es un protocolo sin estado no guarda ninguna informacion sobre conexiones anteriores al finalizar la transaccion todos los datos se pierden por esto se populariz.....
2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....
3. http hypertext transfer protocol protocolo nativo del web https hypertext transfer protocol secure ftps file transfer protocol secure protocolo de transferencia de archivos wss websocket permite una c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and accurate answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 which explains HTTP's purpose in client-server communication, and node 2 which discusses HTTP methods, are ranked higher than irrelevant nodes, like node 3 which only lists various protocols without explaining HTTP's purpose.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions key concepts like 'servidores y clientes web' and 'protocolo HTTP', resulting in a flawless match.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context is mostly filled with irrelevant information about cookies, browsers, and other protocols, as stated in the reasons for irrelevancy, whereas only a few statements like 'el propósito principal del protocolo HTTP en la comunicación entre clientes y servidores web' and 'servidores y clientes web http y https es un protocolo sin estado...' are actually relevant to the input.



## Caso de Prueba: 16 - Fallo

### Input del Usuario:

¿Cuáles son las prácticas ágiles clave para implementar un proyecto de software de manera efectiva?

### Respuesta Generada:

De acuerdo con los documentos proporcionados, las prácticas ágiles clave para implementar un proyecto de software de manera efectiva son: 1. **\*\*Trabajo en equipo\*\***: Fomentar la colaboración entre equipos multidisciplinarios, incluyendo desarrolladores, analistas, diseñadores, testers y otros profesionales para lograr los objetivos del proyecto. 2. **\*\*Gestión de proyectos\*\***: Aplicar técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que se entregue a tiempo y dentro del presupuesto. 3. **\*\*Ciclo de vida del desarrollo de software\*\***: Seguir un proceso estructurado para guiar el desarrollo del software desde la concepción hasta la implementación y el mantenimiento, incluyendo fases como planificación, análisis, diseño, codificación, pruebas y despliegue. 4. **\*\*Metodologías y prácticas ágiles\*\***: Emplear diversas metodologías ágiles, como Scrum, y prácticas como el control de versiones y la integración continua, para gestionar el desarrollo de software de manera efectiva. 5. **\*\*Iteraciones y sprints\*\***: Dividir el proyecto en iteraciones y sprints para trabajar de manera incremental y obtener retroalimentación constante del cliente y los miembros del equipo. 6. **\*\*Colaboración con el cliente\*\***: Mantener una estrecha relación con el cliente para entender sus necesidades y expectativas y asegurarse de que el producto final cumpla con los objetivos del proyecto. 7. **\*\*Respuesta al cambio\*\***: Estar dispuesto a adaptarse a los cambios y ajustes en el proyecto, y ser flexible en la planificación y el desarrollo del software. Estas prácticas ágiles clave permiten a los equipos de desarrollo de software trabajar de manera efectiva, entregando productos de alta calidad y respondiendo a los cambios y necesidades del cliente de manera eficiente.

### Respuesta Esperada:

Las prácticas ágiles clave para implementar un proyecto de software de manera efectiva son: reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas entregas) y retrospectivas.

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas.....

### Contexto Recuperado:

1. metodologia agil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas practicas para trabajar colaborativamente en equipo y obtener el mejor resul.....

2. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

3. interfaces de usuario la interfaz de usuario ui en una aplicacion es el punto de interaccion visual a traves del cual los usuarios se conectan y utilizan las funcionalidades del software incluye todos.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because while the output addresses the key agile practices, it includes some irrelevant statements that are results of implementing agile practices rather than the practices themselves, preventing a perfect score.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the second node, which mentions 'metodologias y practicas' relevant to the topic, is ranked higher than the first and third nodes, which are irrelevant nodes that discuss Scrum methodology and user interfaces respectively, but the irrelevant nodes are not ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.26 (Threshold: 0.5) - The score is 0.26 because most of the retrieval context statements (12 out of 15) do not mention agile practices, and instead focus on general software engineering concepts, user interfaces, testing, and project management, as stated in the reasons for irrelevancy, such as 'The statement does not mention agile practices, it's a general concept of software engineering.' and 'The statement is about user interfaces in software applications, which is not related to agile practices for implementing a software project.'

## Caso de Prueba: 17 - Fallo

### Input del Usuario:

¿Cuáles son las prácticas ágiles más comunes utilizadas en el desarrollo de software y cómo se implementan en un proyecto?

### Respuesta Generada:

Según el Documento 4, las prácticas ágiles más comunes utilizadas en el desarrollo de software son:

1. Reunión diaria: se trata de una reunión breve en la que los miembros del equipo comparten su progreso y planes para el día. 2. Iteraciones cortas: se dividen el trabajo en iteraciones cortas y focalizadas para entregar pequeñas partes funcionales del proyecto. 3. Planeación de la iteración: se planifica el trabajo a realizar en cada iteración, estableciendo prioridades y objetivos claros. 4. Pruebas unitarias: se realizan pruebas individuales de cada unidad de código para asegurar que funcione correctamente. 5. Incrementos de software pequeños: se entrega el software en pequeños incrementos, permitiendo una retroalimentación temprana y continua. 6. Retrospectivas: se realizan reuniones para reflexionar sobre el proceso y identificar oportunidades de mejora. Para implementar estas prácticas en un proyecto, se pueden seguir los siguientes pasos: 1. Establecer un equipo de trabajo que incluya a todos los miembros necesarios para el proyecto. 2. Definir los roles y responsabilidades de cada miembro del equipo. 3. Establecer un marco de tiempo para cada iteración, y planificar el trabajo a realizar en cada una. 4. Realizar reuniones diarias para sincronizar el progreso y planeación del trabajo. 5. Implementar pruebas unitarias y de integración para asegurar la calidad del software. 6. Entregar el software en pequeños incrementos, recopilando retroalimentación y adaptándose a los cambios. 7. Realizar retrospectivas periódicas para evaluar el proceso y identificar oportunidades de mejora. Es importante destacar que cada equipo y proyecto es único, por lo que es fundamental adaptar y personalizar estas prácticas ágiles según las necesidades específicas del proyecto.

### Respuesta Esperada:

Las prácticas ágiles más comunes utilizadas en el desarrollo de software son: \* Reunión diaria \* Iteraciones cortas \* Planeación de la iteración \* Pruebas unitarias \* Incrementos de software (pequeñas entregas) \* Retrospectivas Estas prácticas se implementan en un proyecto de la siguiente manera: \* La reunión diaria se utiliza para sincronizar el trabajo del equipo y establecer objetivos para el día. \* Las iteraciones cortas permiten dividir el proyecto en pequeñas partes manejables y entregar resultados rápidamente. \* La planeación de la iteración se utiliza para definir los objetivos y tareas para cada iteración. \* Las pruebas unitarias se utilizan para asegurar que cada componente del software funcione correctamente. \* Los incrementos de software permiten entregar pequeñas partes del software funcionales y recibir retroalimentación temprana. \* Las retrospectivas se utilizan para reflexionar sobre el proceso y mejorar la eficiencia del equipo. Estas prácticas ágiles se integran en el desarrollo de software a través de la implementación de pruebas unitarias, pruebas funcionales y pruebas End-to-End, y la construcción de pruebas para los diferentes componentes del proyecto.

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas.....

## Contexto Recuperado:

1. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....
2. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....
3. metodologia agil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas practicas para trabajar colaborativamente en equipo y obtener el mejor resul.....

## Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a comprehensive and accurate response without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node provides a general overview of agile software development, and the third-ranked node explains the Scrum framework, which are both highly relevant to the input. However, the second-ranked node, which focuses on estimation of time and cost in agile methodologies, is an irrelevant node that should be ranked lower.

Contextual Recall: 0.08 (Threshold: 0.5) - The score is 0.08 because only sentence 1 in the expected output can be attributed to node 3 in the retrieval context, while the rest of the sentences lack matching information in the retrieval context.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context had many irrelevant statements about costos, administrative tasks, infrastructure, marketing, and insurance, which are not related to common agile practices in software development, as stated in the input, but had some relevant statements about agile development methodologies and Scrum, which boosted the score.

## Caso de Prueba: 18 - Éxito

### Input del Usuario:

¿Cuáles son las mejores prácticas para diseñar APIs ágiles que permitan iteraciones cortas y entregas frecuentes de software?

### Respuesta Generada:

Según los conceptos y principios de desarrollo de software ágil, las mejores prácticas para diseñar APIs ágiles que permitan iteraciones cortas y entregas frecuentes de software son:

1. **Definir y priorizar historias de usuario**: Establecer requerimientos claros y concisos a través de historias de usuario que describan las necesidades del usuario y permitan conocer la base de las necesidades del usuario.
2. **Dividir en tareas técnicas**: Dividir las historias de usuario en tareas técnicas implementables y realizables en iteraciones cortas.
3. **Implementar APIs iterativamente**: Implementar APIs de manera iterativa, comenzando con una API básica y agregando funcionalidades en cada iteración.
4. **Definir criterios de aceptación**: Establecer criterios de aceptación claros y específicos para cada historia de usuario y cada tarea técnica.
5. **Realizar pruebas unitarias y funcionales**: Realizar pruebas unitarias y funcionales para asegurarse de que la API cumpla con los requisitos y criterios de aceptación establecidos.
6. **Revisar y refactorizar código**: Revisar y refactorizar el código en cada iteración para asegurarse de que se cumplan las mejores prácticas de codificación y sea fácil de mantener.
7. **Utilizar APIs RESTful**: Utilizar APIs RESTful para permitir una comunicación flexible y escalable entre sistemas.
8. **Documentar la API**: Documentar la API de manera clara y concisa para que sean entendibles para los desarrolladores y los usuarios.
9. **Realizar retrospectivas**: Realizar retrospectivas en cada iteración para analizar qué funcionó bien, qué no funcionó y qué se puede mejorar en la siguiente iteración.
10. **Asegurarse de la compatibilidad**: Asegurarse de que la API sea compatible con diferentes sistemas y tecnologías.

Siguiendo estas prácticas, se puede diseñar y desarrollar APIs ágiles que permitan iteraciones cortas y entregas frecuentes de software, lo que se traduce en una mayor velocidad y flexibilidad en el desarrollo de aplicaciones.

### Respuesta Esperada:

Para diseñar APIs ágiles que permitan iteraciones cortas y entregas frecuentes de software, se recomiendan las siguientes prácticas:

1. Uso de reuniones diarias para mantener a todos los miembros del equipo informados y alineados con los objetivos del proyecto.
2. Planeación de la iteración para definir los objetivos y priorizar las tareas para cada iteración.
3. Desarrollo de incrementos de software pequeños y frecuentes para permitir la retroalimentación temprana y la adaptación a los cambios.
4. Realización de pruebas unitarias y funcionales para asegurar la calidad del código y detectar errores tempranamente.
5. Implementación de un diseño de API que permita la flexibilidad y la escalabilidad para adaptarse a los cambios y crecer con el proyecto.
6. Mapeo de los diseños para codificación según la arquitectura definida para asegurar la coherencia y la integridad del proyecto.
7. Realización de retrospectivas para reflexionar sobre el proceso y identificar oportunidades de mejora.

Al seguir estas prácticas, se puede lograr un desarrollo de software ágil y flexible que permita iteraciones cortas y entregas frecuentes.

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software)

(pequeñas.....

### **Contexto Recuperado:**

1. aspectos a tratar practicas para el agilismo uso de practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y ret.....
2. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....
3. practicas agiles en scrum dod definition of done criterio cumplido criterios de aceptacion cumplidos la historia de usuario cumple con todos los criterios de aceptacion definidos previamente codigo de.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node mentioning key agile practices, the second node discussing user stories and agile requirements, and the third node providing specific Scrum criteria, resulting in a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output, such as sentences 1, 2, 3, 4, 5, 6, and 7, have direct matches with nodes in the retrieval context, specifically the 1st node, indicating a perfect alignment between the expected output and the retrieval context.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context only partially addresses the input question, with most statements being irrelevant to designing APIs for short iterations and frequent software deliveries, as they focus on agile methodologies, user stories, and Scrum practices, whereas only a few statements, such as 'uso de practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y retrospectivas' and 'implementacion diseno de apis mapeo de los disenos para codificacion segun la arquitectura definida', are directly related to the input.

## Caso de Prueba: 19 - Fallo

### Input del Usuario:

¿Cuáles son las pruebas de software más adecuadas para implementar en un proyecto de desarrollo ágil?

### Respuesta Generada:

Según el Documento 3, en el desarrollo de software ágil, se recomiendan realizar pruebas unitarias, pruebas funcionales y pruebas end-to-end. Estas pruebas son fundamentales para evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer. Las pruebas unitarias se enfocan en evaluar individualmente las unidades de código, garantizando que cada parte del sistema funcione correctamente. Las pruebas funcionales, por otro lado, evalúan la funcionalidad de un sistema o aplicación desde la perspectiva del usuario, comprobando que se cumplan los requisitos y comportamientos esperados. Finalmente, las pruebas end-to-end evalúan el sistema o aplicación en su conjunto, simulando el flujo de usuario completo, desde la entrada de datos hasta la obtención de resultados. En un proyecto de desarrollo ágil, es importante implementar estas pruebas de manera incremental y continua, asegurando que el sistema o aplicación cumpla con los requisitos y sea escalable, mantenible y confiable.

### Respuesta Esperada:

Según el contexto, las pruebas de software más adecuadas para implementar en un proyecto de desarrollo ágil son: \* Pruebas unitarias \* Pruebas funcionales \* Pruebas End-to-End

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas.....

### Contexto Recuperado:

1. metodología agil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas practicas para trabajar colaborativamente en equipo y obtener el mejor resul.....
2. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....
3. aspectos a tratar practicas para el agilismo uso de practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y ret.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because two out of three nodes in the retrieval contexts are ranked higher than they should be, specifically nodes 1 and 2, which do not mention specific software tests suitable for an agile development project, unlike node 3 which explicitly mentions relevant concepts.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output perfectly match the information provided in the 3rd node in retrieval context.

Contextual Relevancy: 0.14 (Threshold: 0.5) - The score is 0.14 because the retrieval context mostly talks about agile methodology, Scrum practices, and software development, but hardly mentions software testing, with only a few relevant statements like 'pruebas unitarias', 'pruebas funcionales', and 'pruebas endtoend'.



## Caso de Prueba: 20 - Fallo

### Input del Usuario:

¿Cuáles son las prácticas ágiles que se pueden implementar para mejorar la estimación de tiempos y costos en el desarrollo de una aplicación de software?

### Respuesta Generada:

Según los documentos proporcionados, algunas prácticas ágiles que se pueden implementar para mejorar la estimación de tiempos y costos en el desarrollo de una aplicación de software son:

1. **\*\*Reuniones diarias\*\*** (Daily Meetings): Para mantener informados a todos los miembros del equipo sobre el progreso del proyecto y detectar posibles problemas que puedan afectar la estimación de tiempos y costos.
2. **\*\*Iteraciones cortas\*\*** (Sprints): Dividir el proyecto en iteraciones cortas para poder estimar y planificar mejor los tiempos y costos de cada iteración.
3. **\*\*Planeación de la iteración\*\*** (Iteration Planning): Realizar una planeación detallada de cada iteración para determinar qué tareas se deben realizar y qué recursos se necesitan.
4. **\*\*Pruebas unitarias\*\***: Realizar pruebas unitarias para detectar errores tempranamente y evitar costos adicionales por correcciones.
5. **\*\*Incrementos de software pequeños\*\*** (Small Increments): Dividir el proyecto en pequeños incrementos para poder entregar algo de valor al usuario y recibir retroalimentación temprana.
6. **\*\*Retrospectivas\*\*** (Retrospectives): Realizar retrospectivas al final de cada iteración para analizar qué se hizo bien y qué se puede mejorar en la próxima iteración.
7. **\*\*Descomponer historias de usuario en tareas\*\*** (User Story Breakdown): Descomponer las historias de usuario en tareas específicas para hacer una estimación más precisa de los tiempos y costos.
8. **\*\*Estimación basada en analogía\*\*** (Analogy-Based Estimation): Utilizar la experiencia pasada para hacer estimaciones de costos y tiempos basadas en proyectos similares.
9. **\*\*Uso de prácticas de estimación de costos\*\*** (Cost Estimation Techniques): Utilizar técnicas de estimación de costos como la estimación basada en puntos de historia (Story Points) o la estimación basada en horas (Hour-Based Estimation).
10. **\*\*Revisión y ajuste continuo\*\*** (Continuous Review and Adjustment): Revisar y ajustar continuamente la estimación de tiempos y costos según se vayan realizando las iteraciones y se obtenga más información sobre el proyecto.

Estas prácticas ágiles pueden ayudar a mejorar la estimación de tiempos y costos en el desarrollo de una aplicación de software, permitiendo una mayor precisión y flexibilidad en la planificación y seguimiento del proyecto.

### Respuesta Esperada:

Las prácticas ágiles que se pueden implementar para mejorar la estimación de tiempos y costos en el desarrollo de una aplicación de software son: \* Reunión diaria \* Iteraciones cortas \* Planeación de la iteración \* Pruebas unitarias \* Incrementos de software (pequeñas entregas) \* Retrospectivas. Estas prácticas permiten una mayor flexibilidad y adaptabilidad en el desarrollo, lo que ayuda a mejorar la estimación de tiempos y costos.

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas.....

### Contexto Recuperado:

1. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....
2. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....
3. aspectos a tratar practicas para el agilismo uso de practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y ret.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval context provides a detailed explanation of estimation of times and costs in agile methodologies, which is highly relevant to the input, and the third-ranked node lists agile practices that can improve the estimation of times and costs in software development. However, the second-ranked node is an irrelevant node that should be ranked lower as it's not directly related to the estimation of times and costs in software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output can be attributed to nodes in the retrieval context, with multiple matches found in the 3rd node, and the concept of 'estimación de tiempos y costos' is present in the 1st node, resulting in a perfect contextual recall.

Contextual Relevancy: 0.47 (Threshold: 0.5) - The score is 0.47 because the retrieval context mostly contains irrelevant information about indirect costs, API design, software testing, deployment strategies, and emerging trends, with only a few statements mentioning agile practices and time/cost estimation, such as 'estimacion de tiempos y costos en marco de las metodologias agiles' and 'practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y retrospectivas'.

## Caso de Prueba: 21 - Fallo

### Input del Usuario:

¿Cuáles son las prácticas ágiles más importantes para implementar en un proyecto de software para asegurar un despliegue exitoso?

### Respuesta Generada:

Basándome en el contexto proporcionado, puedo identificar las prácticas ágiles más importantes para implementar en un proyecto de software para asegurar un despliegue exitoso. A continuación, se presentan algunas de las prácticas ágiles clave mencionadas en los documentos proporcionados: 1. **\*\*Colaboración en equipo\*\***: La metodología ágil de desarrollo de software Scrum se enfoca en trabajar colaborativamente en equipo, como se menciona en el Documento 1. Esto implica que todos los miembros del equipo deben trabajar juntos para lograr el objetivo del proyecto. 2. **\*\*Definición de roles y responsabilidades\*\***: Es importante definir los roles y responsabilidades de cada miembro del equipo, como se menciona en el Documento 1. Esto ayudará a evitar confusiones y asegurará que cada tarea se asigne a la persona adecuada. 3. **\*\*Iteraciones (Sprints)\*\***: La metodología Scrum se divide en iteraciones o Sprints, como se menciona en el Documento 1. Esto permite dividir el proyecto en tareas más pequeñas y manejables, y evaluar el progreso después de cada iteración. 4. **\*\*Estimación de tiempos y costos\*\***: Es fundamental realizar una estimación precisa de los tiempos y costos involucrados en el proyecto, como se menciona en los Documentos 2 y 8. Esto ayudará a planificar adecuadamente el proyecto y a evitar sobrecostos. 5. **\*\*Entrevistas con los usuarios\*\***: Es importante realizar entrevistas con los usuarios y partes interesadas para entender sus necesidades y requerimientos, como se menciona en el Documento 6. Esto ayudará a crear un software que se adapte a las necesidades del cliente. 6. **\*\*Redacción de requerimientos\*\***: Es fundamental redactar los requerimientos de manera clara y concisa, como se menciona en el Documento 7. Esto ayudará a evitar malentendidos y asegurará que el equipo de desarrollo entienda correctamente los requisitos del proyecto. 7. **\*\*Adaptación al cambio\*\***: La mentalidad ágil se enfoca en la respuesta al cambio, como se menciona en el Documento 9. Esto implica que el equipo debe estar dispuesto a adaptarse a los cambios y requerimientos cambiantes durante el desarrollo del proyecto. En resumen, para asegurar un despliegue exitoso en un proyecto de software, es fundamental implementar prácticas ágiles como la colaboración en equipo, la definición de roles y responsabilidades, la estimación de tiempos y costos, las entrevistas con los usuarios, la redacción de requerimientos y la adaptación al cambio.

### Respuesta Esperada:

Las prácticas ágiles más importantes para implementar en un proyecto de software para asegurar un despliegue exitoso son: \* Reunión diaria \* Iteraciones cortas \* Planeación de la iteración \* Pruebas unitarias \* Incrementos de software (pequeñas entregas) \* Retrospectivas Estas prácticas permiten un desarrollo más flexible y adaptable, lo que reduce el riesgo de errores y permite una entrega más rápida y segura del software.

### Contexto Relevante:

1. Aspectos a tratar Prácticas para el agilismo • Uso de prácticas ágiles como (reunión diaria, iteraciones cortas, planeación de la iteración, pruebas unitarias, incrementos de software (pequeñas.....

### **Contexto Recuperado:**

1. metodologia agil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas practicas para trabajar colaborativamente en equipo y obtener el mejor resul.....
2. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....
3. objetivos del curso objetivos especificos entender como estimar un producto de software tiempo y valor en de una aplicacion de software desarrollar una aplicacion de software aplicar algunos modelos .....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which mentions 'scrum' and its phases, are ranked higher than the irrelevant nodes, like the second and third nodes which talk about estimation of times and costs and course objectives respectively, ensuring a perfect ranking.

Contextual Recall: 0.14 (Threshold: 0.5) - The score is 0.14 because only the 1st sentence in the expected output, 'Reunión diaria', can be attributed to the 1st node in the retrieval context, which mentions 'metodologia agil de desarrollo de software scrum', while the rest of the sentences lack supporting evidence in the retrieval context.

Contextual Relevancy: 0.42 (Threshold: 0.5) - The score is 0.42 because the retrieval context mostly contains irrelevant information about roles, sprint iteration, tools, and software development, with only a few statements mentioning agile methodologies and practices, such as 'metodologia agil de desarrollo de software scrum' and 'estimacion de tiempos y costos en marco de las metodologias agiles', which are somewhat related to the input.

## Caso de Prueba: 22 - Éxito

### Input del Usuario:

¿Cuáles son los aspectos clave que justifican la importancia de estudiar Ingeniería de Software en la actualidad?

### Respuesta Generada:

Según los documentos proporcionados, los aspectos clave que justifican la importancia de estudiar Ingeniería de Software en la actualidad son: 1. **Creciente demanda de desarrolladores de software con talento**: La tecnología avanza a un ritmo acelerado, lo que hace que la demanda de desarrolladores de software con habilidades y conocimientos en Ingeniería de Software sea cada vez más alta. 2. **Disciplina madura y necesaria**: La Ingeniería de Software se ha vuelto cada vez más importante en nuestra vida cotidiana, lo que la hace una disciplina madura y necesaria en la actualidad. 3. **Reconocimiento internacional**: La Ingeniería de Software es una profesión de clase mundial, y en un futuro dominado por la Inteligencia Artificial, los ingenieros de software asumirán roles más estratégicos y especializados, como diseñadores de sistemas, auditores éticos, integradores tecnológicos y supervisores de IA. 4. **Capacidad de crear software de calidad**: La Ingeniería de Software busca la creación de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios, lo que es crítico en la actualidad debido a la creciente dependencia en el software en nuestra vida cotidiana. 5. **Importancia en la creación de aplicaciones de software**: La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad, lo que es fundamental en la actualidad debido a la creciente demanda de software en diferentes áreas. En resumen, la importancia de estudiar Ingeniería de Software en la actualidad se justifica por la creciente demanda de desarrolladores de software con talento, la madurez y necesidad de la disciplina, el reconocimiento internacional, la capacidad de crear software de calidad y la importancia en la creación de aplicaciones de software.

### Respuesta Esperada:

Los aspectos clave que justifican la importancia de estudiar Ingeniería de Software en la actualidad son: \* La aplicación del conocimiento y comprensión teórica para crear aplicaciones de software de calidad. \* La alta demanda de desarrolladores de software con talento, ya que la tecnología avanza y se busca la capacidad de crear software de calidad considerando aspectos como el diseño, el desarrollo, la integración y la ética. \* La madurez de la disciplina, que se vuelve cada vez más importante en nuestra vida cotidiana. \* El reconocimiento internacional de la profesión, que asumirá roles más estratégicos y especializados en un futuro dominado por la inteligencia artificial.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se

e.....

2. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l ta d e m.....

3. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al disenio desarrollo implementacion pruebas desplieg.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.93 (Threshold: 0.5) - The score is 0.93 because the response is highly relevant, but slightly derailed by an unnecessary mention of recognition, which doesn't directly justify the importance of studying Software Engineering.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked from 1 to 3, are highly relevant to the input, with the first node explaining the application of engineering principles, the second node supporting the application of theoretical knowledge, and the third node providing a comprehensive understanding of software engineering, making it a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 2nd node in the retrieval context, demonstrating a seamless connection between the output and the context.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the input, providing a comprehensive overview of the key aspects that justify the importance of studying Software Engineering, covering concepts, principles, methodologies, and practices that are directly relevant to the field.

## Caso de Prueba: 23 - Éxito

### Input del Usuario:

¿Cuáles son las razones clave para estudiar ingeniería de software en la actualidad?

### Respuesta Generada:

Según el Documento 3 y el Documento 5, las razones clave para estudiar Ingeniería de Software en la actualidad son: 1. **\*\*Necesidad creciente de desarrolladores de software con talento\*\***: A medida que avanza la tecnología, se busca la capacidad de crear software de calidad. 2. La **\*\*capacidad de crear software de calidad\*\*** que satisfaga las necesidades y expectativas de los usuarios, teniendo en cuenta el diseño, el desarrollo, la integración y aspectos éticos, entre otros. 3. La **\*\*importancia creciente del software en nuestra vida cotidiana\*\***: La Ingeniería de Software se vuelve cada vez más importante en nuestra vida cotidiana. 4. **\*\*Reconocimiento internacional\*\***: La Ingeniería de Software es una profesión de clase mundial. 5. **\*\*Oportunidades laborales estratégicas\*\***: En un futuro dominado por la Inteligencia Artificial, los ingenieros de software asumirán roles más estratégicos y especializados, como diseñadores de sistemas, auditores éticos, integradores tecnológicos y supervisores de IA. En resumen, la demanda de desarrolladores de software con habilidades y conocimientos en Ingeniería de Software es alta, y la importancia de esta profesión en la sociedad y la economía es cada vez más relevante.

### Respuesta Esperada:

Las razones clave para estudiar ingeniería de software en la actualidad son: la aplicación del conocimiento y comprensión teórica para la creación de aplicaciones de software de calidad, la alta demanda de desarrolladores de software con talento, la importancia creciente de la disciplina en nuestra vida cotidiana, y el reconocimiento internacional de la profesión, que asumirá roles más estratégicos y especializados en un futuro dominado por la inteligencia artificial.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....
2. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l t a d e m.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.88 (Threshold: 0.5) - The score is 0.88 because the response is mostly relevant to the input, but slightly derailed by an off-topic statement about software quality, preventing a perfect score.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which provides key concepts of software engineering, is correctly ranked high, and the third-ranked node, which directly addresses the question, is also ranked correctly. However, the second-ranked node, which focuses on obtaining non-functional requirements, is an irrelevant node that should be ranked lower, bringing the score down.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which provides a comprehensive match for all the mentioned reasons.

Contextual Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the retrieval context has many irrelevant statements, such as 'The statement is about conducting an interview, not about studying software engineering' and 'The statement is about identifying potential interviewees, not about studying software engineering.', but it also has some relevant statements like 'conceptos aspectos clave de la ingeniería de software' and 'aplicación de principios de ingeniería la ingeniería de software utiliza conceptos de ingeniería como el análisis de requisitos el diseño la construcción las pruebas y el mantenimiento para garantizar que el software sea confiable eficiente y escalable' that are related to studying software engineering.



## Caso de Prueba: 24 - Éxito

### Input del Usuario:

¿Cuáles son las razones clave para estudiar ingeniería de software en la actualidad?

### Respuesta Generada:

Según los documentos proporcionados, las razones clave para estudiar ingeniería de software en la actualidad son: 1. **\*\*La creciente demanda de desarrolladores de software con talento\*\***: A medida que avanza la tecnología, se busca la capacidad de crear software de calidad, lo que genera una demanda creciente de profesionales capacitados en ingeniería de software. 2. **\*\*La importancia de la ingeniería de software en nuestra vida cotidiana\*\***: La ingeniería de software se vuelve cada vez más importante en nuestra vida cotidiana, lo que hace que sea una disciplina valiosa y necesaria para abordar los desafíos tecnológicos actuales. 3. **\*\*El reconocimiento internacional de la profesión\*\***: La ingeniería de software es una profesión de clase mundial, y los ingenieros de software asumirán roles más estratégicos y especializados en un futuro dominado por la inteligencia artificial. En resumen, estudiar ingeniería de software es relevante en la actualidad debido a la demanda creciente de profesionales capacitados, la importancia de la disciplina en nuestra vida cotidiana y el reconocimiento internacional de la profesión.

### Respuesta Esperada:

Las razones clave para estudiar ingeniería de software en la actualidad son: 1. Aplicar conocimiento: La ingeniería de software te permite aplicar el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. 2. Alta demanda: Existe una necesidad creciente de desarrolladores de software con talento, ya que se busca la capacidad de crear software de calidad teniendo en cuenta el diseño, el desarrollo, la integración, aspectos éticos, entre otros. 3. Disciplina madura: La ingeniería de software es una disciplina que madura y se vuelve cada vez más importante en nuestra vida cotidiana. 4. Reconocimiento internacional: Es una profesión de clase mundial y, en un futuro dominado por la IA, los ingenieros de software asumirán roles más estratégicos y especializados.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

2. requerimientos no funcionales técnicas para obtener requerimientos la entrevista es un diálogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l t a d e m.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is highly relevant, but slightly derailed by a general statement about technology advancing, which could be more focused on software engineering.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which explains the application of knowledge in software engineering, is correctly ranked higher than the second node, which talks about non-functional requirements and is not directly related to the reasons for studying software engineering, but the irrelevant node at rank 2 should be ranked lower than the relevant node at rank 3, which mentions the high demand for talented software developers, the maturity of the discipline, and international recognition.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output can be perfectly attributed to the 3rd node in the retrieval context, showcasing a flawless alignment between the two.

Contextual Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the retrieval context has some relevant statements about software engineering, such as 'conceptos aspectos clave de la ingenieria de software' and 'aplicacion de principios de ingenieria', but is largely irrelevant due to the presence of many unrelated statements, as mentioned in the reasons for irrelevancy.

## Caso de Prueba: 25 - Éxito

### Input del Usuario:

¿Cuál es el papel de la integración de sistemas en la creación de aplicaciones de software de calidad?

### Respuesta Generada:

Según los documentos proporcionados, la integración de sistemas juega un papel crucial en la creación de aplicaciones de software de calidad. En particular, la arquitectura de aplicaciones web (Documento 6) se refiere a la forma de organizar el código de la aplicación, diseño y actividades de desarrollo y documentación, lo que implica la integración de múltiples tecnologías funcionando simultáneamente. Además, la documentación también destaca la importancia de la arquitectura desacoplada con API (Documento 3), que separa los componentes de una aplicación para que funcionen de manera independiente, comunicándose entre sí a través de interfaces estandarizadas (APIs). Esto facilita la escalabilidad, el mantenimiento y la integración con otros sistemas. En este sentido, la integración de sistemas es fundamental para crear aplicaciones de software de calidad, ya que permite la comunicación efectiva y la coordinación entre los diferentes componentes de la aplicación, lo que a su vez permite la creación de software de alta calidad que satisface las necesidades y expectativas de los usuarios. En resumen, la integración de sistemas es un papel clave en la creación de aplicaciones de software de calidad, ya que facilita la interoperabilidad, la escalabilidad y el mantenimiento de los sistemas, lo que es esencial para la creación de software de alta calidad.

### Respuesta Esperada:

El papel de la integración de sistemas en la creación de aplicaciones de software de calidad es fundamental, ya que se busca crear software que tenga en cuenta el diseño, el desarrollo, la integración, aspectos éticos, entre otros. La integración de sistemas es un aspecto clave en la creación de software de calidad, ya que permite combinar diferentes componentes y sistemas para lograr un objetivo común.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue.....

2. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de prácticas principios y procedimientos que guían y regulan el proceso de creación.....

3. interfaz de usuario lógica de la aplicación interfaz de datos API frontend backend arquitectura para aplicaciones desacopladas basadas en API BDatos web service servidor web tipos de estilos

arquitect.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the response provided some relevant information, but was heavily diluted by statements about application architecture that didn't directly address the role of system integration in creating quality software applications.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the nodes in the retrieval contexts that directly address the role of system integration in creating high-quality software, such as node 1 mentioning software engineering principles and node 2 highlighting methodologies in software creation, are ranked higher than the irrelevant node 3, which focuses on application architecture.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, with node 1 mentioning 'creacion de software de alta calidad' and node 2 discussing 'integracion de sistemas' and 'combinar diferentes componentes y sistemas', making it a seamless match.

Contextual Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the retrieval context, although containing some relevant statements like 'la ingenieria de software se enfoca en la creacion sistematica y eficiente de software de alta calidad' and 'una arquitectura desacoplada con api separa los componentes de una aplicacion para que funcionen de manera independiente', is largely irrelevant due to the presence of overly broad information like 'interfaz de usuario logica de la aplicacion interfaz de datos api frontend backend arquitectura para aplicaciones desacopladas basadas en api bdatos web service servidor web tipos de estilos arquitecturales para las aplicaciones web'.

## Caso de Prueba: 26 - Éxito

### Input del Usuario:

¿Cuál es el papel de la ética en el desarrollo de software de calidad y cómo los ingenieros de software pueden asegurar que sus creaciones tengan un impacto positivo en la sociedad?

### Respuesta Generada:

Según el contexto proporcionado, la ética juega un papel crucial en el desarrollo de software de calidad. La ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios (Documento 2). Sin embargo, este enfoque no solo se centra en la creación de software funcional, sino que también considera aspectos éticos (Documento 1 y 4). Los ingenieros de software deben asegurarse de que sus creaciones tengan un impacto positivo en la sociedad. Para lograr esto, deben considerar la ética en cada fase del desarrollo de software, desde la recolección de requisitos hasta el despliegue y soporte. Algunos aspectos éticos que deben ser considerados incluyen: 1. Privacidad y seguridad: Los ingenieros de software deben diseñar sistemas que protejan la información de los usuarios y minimicen los riesgos de vulnerabilidades. 2. Accesibilidad: Debe asegurarse de que el software sea accesible para todos los usuarios, incluyendo aquellos con discapacidades. 3. Transparencia: Los ingenieros de software deben ser transparentes en cuanto a cómo se recopila, almacena y utiliza la información de los usuarios. 4. Responsabilidad: Deben asumir la responsabilidad de sus creaciones y garantizar que no causen daños a los usuarios o la sociedad en general. Para asegurar que sus creaciones tengan un impacto positivo, los ingenieros de software pueden adoptar prácticas éticas en su trabajo, como: 1. Realizar análisis de impacto ético en cada fase del desarrollo de software. 2. Incorporar la ética en la toma de decisiones y en la definición de los requisitos del proyecto. 3. Colaborar con otros profesionales y expertos en ética para obtener una perspectiva más amplia. 4. Establecer código de conducta y políticas éticas claras en la organización. En resumen, la ética es fundamental en el desarrollo de software de calidad. Los ingenieros de software deben considerar la ética en cada fase del desarrollo y adoptar prácticas éticas para asegurar que sus creaciones tengan un impacto positivo en la sociedad.

### Respuesta Esperada:

Según el contexto, la ética juega un papel importante en el desarrollo de software de calidad, ya que se menciona como uno de los aspectos a tener en cuenta en la creación de aplicaciones de software. Los ingenieros de software pueden asegurar que sus creaciones tengan un impacto positivo en la sociedad al considerar la ética en el diseño, desarrollo e integración de sus aplicaciones. De esta manera, pueden crear software que no solo sea funcional y eficiente, sino también que tenga en cuenta las implicaciones sociales y éticas de su uso.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l ta d e m.....
2. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al disenio desarrollo implementacion pruebas desplieg.....
3. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.93 (Threshold: 0.5) - The score is 0.93 because the response is highly relevant, but slightly loses focus by mentioning the general goal of software engineering, which, although related, is not the primary concern of the input question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 mentioning ethics as an important aspect in software development and node 2 explaining the importance of software engineering in creating high-quality software, are ranked higher than the irrelevant nodes, like node 3 which is about agile development and not directly related to the role of ethics in software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, as node 1 mentions 'aplicaciones de software de calidad' and 'aspectos eticos', which are key concepts in the output sentences.

Contextual Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because the retrieval context has some relevant statements that mention software quality, ethics, and social impact, but is brought down by irrelevant statements that do not address the role of ethics in software development, such as 'reconocimiento internacional es una profesion de clase mundial' and 'por que estudiar ingenieria de software'.

## Caso de Prueba: 27 - Fallo

### Input del Usuario:

¿Cómo puede la Inteligencia Artificial influir en el papel de los ingenieros de software en el futuro?

### Respuesta Generada:

Según el Documento 4, en un futuro dominado por la Inteligencia Artificial (IA), los ingenieros de software asumirán roles más estratégicos y especializados, como diseñadores de sistemas, auditores éticos, integradores tecnológicos y supervisores de IA. Esto sugiere que la IA influirá en el papel de los ingenieros de software, otorgándoles responsabilidades más avanzadas y especializadas. Además, según el Documento 8, la IA puede aumentar la ingeniería de software, automatizando partes del ciclo de vida, como el despliegue, pruebas y parte de la codificación. Esto puede llevar a una mayor eficiencia y productividad en el desarrollo de software, permitiendo a los ingenieros de software enfocarse en tareas más complejas y creativas. En resumen, la IA puede influir en el papel de los ingenieros de software en el futuro, otorgándoles roles más especializados y responsabilidades más avanzadas, y automatizando tareas repetitivas y tediosas, lo que permitirá a los ingenieros de software enfocarse en tareas más complejas y creativas.

### Respuesta Esperada:

Según el contexto, la Inteligencia Artificial (IA) influirá en el papel de los ingenieros de software en el futuro, llevándolos a asumir roles más estratégicos y especializados, como diseñadores de sistemas, auditores éticos, integradores tecnológicos y supervisores de IA.

### Contexto Relevante:

1. APLICAR CONOCIMIENTO La Ingeniería de Software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad. A L T A D E.....

### Contexto Recuperado:

1. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l t a d e m.....
2. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....
3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the most relevant node, ranked 1, directly addressing the question, and the irrelevant nodes, ranked 2 and 3, providing definitions and discussing unrelated topics, respectively.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the future roles of software engineers, resulting in a flawless match.

Contextual Relevancy: 0.18 (Threshold: 0.5) - The score is 0.18 because the retrieval context mostly discusses unrelated topics such as application of knowledge in software engineering, growing need for talented software developers, and user interface prototyping, with only a few statements mentioning the strategic roles of software engineers, like 'diseñadores de sistemas auditores éticos integradores tecnológicos y supervisores de ia', being relevant to the input.



## Caso de Prueba: 28 - Fallo

### Input del Usuario:

¿Cuál es el propósito del atributo "async" en un elemento de script en HTML5?

### Respuesta Generada:

No hay mención a "async" en ninguno de los contextos proporcionados. Sin embargo, puedo responder a la pregunta basándome en conocimientos generales sobre HTML5. El atributo "async" en un elemento de script en HTML5 indica que el script se debe cargar de forma asincrónica. Es decir, el navegador no bloqueará la carga de la página mientras se carga el script. En su lugar, el script se cargará en segundo plano mientras la página continúa cargando. El propósito de "async" es mejorar el rendimiento de la página, ya que permite que el navegador continúe cargando otros recursos mientras se carga el script. Esto es especialmente útil cuando se carga un script grande o remoto que podría bloquear la carga de la página. Al marcar un script como "async", se garantiza que la página se cargue rápidamente y que el script se ejecute en segundo plano.

### Respuesta Esperada:

No se puede determinar el propósito del atributo "async" en un elemento de script en HTML5 con el contexto proporcionado, ya que no se menciona el atributo "async" en el contexto. El contexto parece ser una lista de herramientas y tecnologías relacionadas con el desarrollo web, pero no proporciona información específica sobre el atributo "async".

### Contexto Relevante:

1. Herramientas Algunas herramientas requeridas Herramientas de desarrollo : Front end HTML5 JavaScript y CSS3 Python Django Bases de datos: PostgreSQL, Oracle Herramientas de Gestión (Google Docs, Jira.....

### Contexto Recuperado:

1. arquitectura hexagonal puertos y adaptadores tipos de estilos arquitecturales para las aplicaciones web script archivo logica de la aplicacion la arquitectura hexagonal es un estilo arquitectural de p.....
2. sprint es un intervalo de tiempo prefijado durante el cual se realizan hu y el resultado es un incremento de software potencialmente entregable un sprint inicia con una planeacion de las hu a realiza.....
3. practicas agiles en scrum backlog refinement sprint grooming opcional el definition of ready dor en scrum es un conjunto de criterios que deben cumplirse para que un elemento del product backlog pueda.....

### Métricas de Evaluación:

Answer Relevancy: 0.88 (Threshold: 0.5) - The score is 0.88 because the response mostly addresses the purpose of the 'async' attribute in HTML5, but includes a minor irrelevant statement that detracts from the overall relevance.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the 1st node discussing software architecture, the 2nd node discussing Agile development methodologies, and the 3rd node discussing Scrum practices, are irrelevant to the input and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output, specifically sentences 1, does not match any part of the retrieval context, which discusses unrelated topics like arquitectura hexagonal and practicas agiles in scrum, and does not mention HTML5 or the 'async' attribute.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely irrelevant to the input, as all the statements talk about unrelated topics such as architectural styles, script files, application logic, and agile practices, with no mention of the 'async' attribute in HTML5.

## Caso de Prueba: 29 - Fallo

### Input del Usuario:

¿Cuál es la mejor práctica para depurar y solucionar errores en un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente?

### Respuesta Generada:

La pregunta no se relaciona directamente con ninguno de los documentos proporcionados, ya que no menciona específicamente depurar y solucionar errores en un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente. Sin embargo, podemos inferir algunas buenas prácticas para depurar y solucionar errores en un proyecto de desarrollo de software en general, y adaptarlas a un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente. Una buena práctica para depurar y solucionar errores es utilizar una metodología ágil como Scrum (Documento 2), que enfatiza la colaboración y la iteración para lograr el mejor resultado posible. En la fase de desarrollo, se pueden utilizar técnicas como la retroalimentación durante el diseño (Documento 1) para identificar y solucionar errores lo antes posible. Además, es importante tener una buena arquitectura de software que permita una separación clara de los componentes y la interfaz de usuario, como la arquitectura hexagonal (Documento 9) o la arquitectura de micro frontend (Documento 10). Esto permitirá identificar y solucionar errores de manera más eficiente. En cuanto a la depuración y solución de errores en un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente, algunas buenas prácticas incluyen: \* Utilizar herramientas de depuración like console.log() o herramientas de desarrollo de navegador como la consola de Chrome DevTools. \* Verificar la respuesta del servidor utilizando la consola de red del navegador o herramientas como Postman. \* Utilizar una metodología ágil para iterar y refactorizar el código. \* Realizar pruebas unitarias y de integración para verificar el comportamiento del código. \* Realizar un seguimiento de los errores y solventarlos lo antes posible. Es importante mencionar que no hay una respuesta única para depurar y solucionar errores en un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente, ya que cada proyecto es único y puede requerir enfoques y herramientas específicas. Sin embargo, siguiendo buenas prácticas de desarrollo de software y adaptándolas a las necesidades específicas del proyecto, se puede reducir la probabilidad de errores y mejorar la calidad del software.

### Respuesta Esperada:

Según el contexto, la mejor práctica para depurar y solucionar errores en un proyecto de desarrollo web que utiliza JavaScript en el lado del cliente es utilizar herramientas como SonarLint, que es una herramienta de análisis de código estático que ayuda a detectar y solucionar errores y vulnerabilidades en el código.

### Contexto Relevante:

1. Herramientas Algunas herramientas requeridas Herramientas de desarrollo : Front end HTML5 JavaScript y CSS3 Python Django Bases de datos: PostgreSQL, Oracle Herramientas de Gestión ([Google Docs, Jira.....

### Contexto Recuperado:

1. interfaces de usuario la interfaz de usuario ui en una aplicacion es el punto de interaccion visual a traves del cual los usuarios se conectan y utilizan las funcionalidades del software incluye todos.....
2. metodologia agil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas practicas para trabajar colaborativamente en equipo y obtener el mejor resul.....
3. 1xx mensajes informativos no se utilizan y estan reservados para un futuro uso 3xx mensajes de redireccion que informan de operaciones complementarias que se deben realizar para finalizar la operacion.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.92 (Threshold: 0.5) - The score is 0.92 because the response is highly relevant to the input, but slightly derailed by an unrelated statement, preventing a perfect score.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all the top-ranked nodes in the retrieval contexts, including the 1st node about user interface design, the 2nd node about the Scrum framework, and the 3rd node about HTTP status codes, are irrelevant to the input, and should be ranked lower than relevant nodes.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context, indicating a complete mismatch.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely irrelevant to the input, with all statements talking about unrelated topics such as user interfaces, Scrum methodology, and HTTP status codes, and none of them addressing debugging and solving errors in a web development project that uses JavaScript on the client-side.

## Caso de Prueba: 30 - Fallo

### Input del Usuario:

¿Cuál es la diferencia entre las propiedades "display: inline" y "display: inline-block" en CSS3?

### Respuesta Generada:

Lo siento, pero la pregunta no se relaciona con el contexto proporcionado, que se enfoca en conceptos de desarrollo de software, métodos HTTP, variables de sesión, y arquitectura de aplicaciones web. La pregunta menciona propiedades CSS3, que no se mencionan en ninguno de los documentos proporcionados. Si deseas obtener una respuesta a esta pregunta, te recomiendo buscar información en recursos de CSS3 o formular la pregunta en un foro específico de desarrollo web.

### Respuesta Esperada:

No hay información en el contexto proporcionado que explique la diferencia entre las propiedades "display: inline" y "display: inline-block" en CSS3. El contexto parece ser una lista de herramientas y tecnologías utilizadas en desarrollo web, pero no proporciona información sobre CSS o estilos.

### Contexto Relevante:

1. Herramientas Algunas herramientas requeridas Herramientas de desarrollo : Front end HTML5 JavaScript y CSS3 Python Django Bases de datos: PostgreSQL, Oracle Herramientas de Gestión (Google Docs, Jira.....

### Contexto Recuperado:

1. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....
2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....
3. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

### Métricas de Evaluación:

Answer Relevancy: 0.12 (Threshold: 0.5) - The score is 0.12 because the actual output is mostly irrelevant to the input, failing to address the difference between 'display: inline' and 'display: inline-block' in CSS3.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the first node which discusses HTTP methods, the second node which also discusses HTTP methods, and the third node which talks about non-functional requirements, are irrelevant to the input question about CSS3 properties, and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output, specifically sentences 1, does not relate to any node in the retrieval context, which primarily discusses HTTP methods, get and post, and non-functional requirements.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely unrelated to the input, with all statements talking about HTTP methods or other unrelated topics, as stated in the reasons for irrelevancy, such as 'The statement is about HTTP methods and not about CSS3 properties 'display: inline' and 'display: inline-block'.', and there are no relevant statements in the retrieval context.

## Caso de Prueba: 31 - Fallo

### Input del Usuario:

¿Cuál es el comando para crear un índice en una tabla existente en PostgreSQL?

### Respuesta Generada:

Lo siento, pero en el contexto proporcionado no hay información relacionada con PostgreSQL ni con la creación de índices en bases de datos. Los documentos proporcionados se refieren a conceptos de desarrollo de software, como la planificación de proyectos, la arquitectura de software, la elicitation de requerimientos, la creación de historias de usuario, el ciclo de vida del desarrollo de software, entre otros. Si necesitas ayuda con la creación de índices en PostgreSQL, te puedo proporcionar la siguiente información: El comando para crear un índice en una tabla existente en PostgreSQL es el siguiente: `CREATE INDEX nombre_indice ON tabla_nombre (columna_nombre);` Donde `nombre_indice` es el nombre del índice que se va a crear, `tabla_nombre` es el nombre de la tabla en la que se va a crear el índice, y `columna_nombre` es el nombre de la columna que se va a indexar. Por ejemplo, para crear un índice en una columna llamada `id_usuario` en una tabla llamada `usuarios`, el comando sería: `CREATE INDEX idx_id_usuario ON usuarios (id_usuario);` Espero que esta información sea útil. Si tienes alguna otra pregunta relacionada con desarrollo de software, estoy aquí para ayudarte.

### Respuesta Esperada:

Lo siento, pero en el contexto proporcionado no se menciona el comando para crear un índice en una tabla existente en PostgreSQL. El contexto parece ser una lista de herramientas y tecnologías utilizadas en un proyecto, pero no proporciona información específica sobre comandos de PostgreSQL.

### Contexto Relevante:

1. Herramientas Algunas herramientas requeridas Herramientas de desarrollo : Front end HTML5 JavaScript y CSS3 Python Django Bases de datos: PostgreSQL, Oracle Herramientas de Gestión (Google Docs, Jira.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....
2. arquitectura hexagonal puertos y adaptadores tipos de estilos arquitecturales para las aplicaciones web script archivo logica de la aplicacion la arquitectura hexagonal es un estilo arquitectural de p.....
3. ejercicio elaborar la descripcion de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad el sistema debe permitir crear un usuario del sistema...

### Métricas de Evaluación:

Answer Relevancy: 0.44 (Threshold: 0.5) - The score is 0.44 because the output contains some relevant information, but it's overshadowed by multiple irrelevant statements that don't address the specific question about creating an index in PostgreSQL.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant to the input, with the first node being about project planning and management, the second node being about software architecture, and the third node being about creating a functional requirement for a university course registration system.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely unrelated to the input, with statements discussing project objectives, cost estimation, work plans, and architectural styles, as quoted in the reasons for irrelevancy, with no relevant statements found.



## Caso de Prueba: 32 - Fallo

### Input del Usuario:

¿Cuál es la mejor práctica para implementar la autenticación de usuarios en una aplicación Django utilizando PostgreSQL como base de datos?

### Respuesta Generada:

La pregunta parece estar fuera del contexto proporcionado, ya que los documentos proporcionados no mencionan específicamente la autenticación de usuarios en una aplicación Django utilizando PostgreSQL como base de datos. Sin embargo, puedo proporcionar una respuesta general basada en la experiencia y conocimientos en desarrollo de software. La mejor práctica para implementar la autenticación de usuarios en una aplicación Django utilizando PostgreSQL como base de datos es utilizar el módulo de autenticación built-in de Django, que se encuentra en `django.contrib.auth`. Django proporciona una forma sencilla de autenticar usuarios a través de la función `authenticate()` que verifica la contraseña de un usuario y devuelve un objeto de usuario autenticado si las credenciales son válidas. También proporciona una forma de crear usuarios y asignarles permisos. Para utilizar la autenticación de Django con PostgreSQL, debes crear un modelo de usuario personalizado que herede de `AbstractBaseUser` y `PermissionsMixin`, y luego configurar la base de datos para utilizar PostgreSQL. Algunas buenas prácticas para implementar la autenticación de usuarios en Django incluyen: \* Utilizar hashed passwords para almacenar contraseñas de usuarios de forma segura. \* Utilizar tokens de autenticación para autenticar a los usuarios en lugar de utilizar sesiones. \* Utilizar middleware de autenticación para proteger vistas y URLs que requieren autenticación. \* Utilizar grupos y permisos para controlar el acceso a funcionalidades y recursos. Es importante mencionar que la seguridad es un aspecto crítico en cualquier aplicación, por lo que es fundamental implementar la autenticación de usuarios de forma segura y escalable. En cuanto a los documentos proporcionados, aunque no mencionan específicamente la autenticación de usuarios en Django con PostgreSQL, algunos de ellos mencionan conceptos relacionados con el desarrollo de software, como la creación de interfaces de usuario, la estimación de historias de usuario, la descomposición de historias de usuario en tareas y la arquitectura de aplicaciones desacopladas.

### Respuesta Esperada:

La mejor práctica para implementar la autenticación de usuarios en una aplicación Django utilizando PostgreSQL como base de datos es utilizar el sistema de autenticación integrado de Django, que utiliza una tabla de usuarios en la base de datos para almacenar la información de los usuarios. Django proporciona un modelo de usuario predeterminado que se puede utilizar para almacenar la información de los usuarios, y también proporciona vistas y formularios para manejar el registro, inicio de sesión y cierre de sesión de los usuarios. Para utilizar el sistema de autenticación de Django, debes configurar la base de datos PostgreSQL en el archivo `settings.py` de tu proyecto Django, y luego crear las tablas de la base de datos utilizando el comando `python manage.py migrate`. Luego, puedes utilizar las vistas y formularios de autenticación de Django para manejar el registro y el inicio de sesión de los usuarios. Por ejemplo, puedes utilizar la vista `LoginView` para manejar el inicio de sesión de los usuarios, y la vista `LogoutView` para manejar el cierre de sesión. Además, es importante utilizar SSL/TLS para cifrar la comunicación entre el cliente y el servidor, y utilizar contraseñas hash para almacenar las contraseñas de los usuarios de manera segura. En resumen, la mejor práctica para implementar la autenticación de usuarios en una aplicación Django utilizando PostgreSQL como base de datos es utilizar el sistema de autenticación integrado de Django, configurar la base de datos PostgreSQL, crear las tablas de la base de datos, y utilizar las vistas y formularios de autenticación de Django para manejar el registro y el inicio de sesión de los usuarios.

### **Contexto Relevante:**

1. Herramientas Algunas herramientas requeridas Herramientas de desarrollo : Front end HTML5 JavaScript y CSS3 Python Django Bases de datos: PostgreSQL, Oracle Herramientas de Gestión ([Google Docs, Jira.....

### **Contexto Recuperado:**

1. interfaces de usuario la interfaz de usuario ui en una aplicacion es el punto de interaccion visual a traves del cual los usuarios se conectan y utilizan las funcionalidades del software incluye todos.....
2. interfaces de usuario interfaces de usuario es una secuencia de imagenes o prototipos de las interfaces de una app que muestran el conjunto de pantallas o interfaces de usuario de una aplicacion las c.....
3. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.77 (Threshold: 0.5) - The score is 0.77 because the response is mostly relevant, but has some general statements that don't specifically address the input question about implementing user authentication in a Django application using PostgreSQL as the database.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant, with the first node being about 'interfaces de usuario', the second node again being about 'interfaces de usuario', and the third node being about 'requerimientos en las metodologias agiles', all of which are not related to the topic of user authentication in a Django application using PostgreSQL.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely irrelevant to the input, with all statements discussing user interfaces, UI, and design, which has no relation to implementing user authentication in a Django application using PostgreSQL.

## Caso de Prueba: 33 - Fallo

### Input del Usuario:

¿Cuáles son los componentes que componen el software según la definición?

### Respuesta Generada:

Según el Documento 2, la arquitectura de software se define como "la organización fundamental de un sistema encarnada en sus componentes, sus relaciones entre sí y con el medio en que se desarrolla, así como los principios rectores de su diseño y evolución". En este sentido, los componentes que componen el software son: \* Componentes: Son los elementos básicos que componen el software, que pueden ser programas, módulos, objetos, clases, funciones, variables, etc. \* Relaciones entre sí: Se refiere a cómo los componentes interactúan entre sí para lograr el funcionamiento del sistema. \* Relaciones con el medio en que se desarrolla: Se refiere a cómo el software interactúa con su entorno, es decir, con los usuarios, otros sistemas, hardware, etc. En resumen, los componentes que componen el software son los elementos básicos del sistema, sus relaciones entre sí y su interacción con el medio en que se desarrolla.

### Respuesta Esperada:

Según la definición, los componentes que componen el software son: I. Instrucciones (programas de computador) D. Documentos que describen la operación y uso de los programas E. Estructuras de datos que permiten a los programas manipular la información adecuadamente.

### Contexto Relevante:

1. Conceptos ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un sistema informático (hardware), le permite.....

### Contexto Recuperado:

1. aspectos a tratar practicas para el agilismo uso de practicas agiles como reunion diaria iteraciones cortas planeacion de la iteracion pruebas unitarias incrementos de software pequenas entregas y ret.....
2. de software arquitectura a b c una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla asi como los princ.....
3. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy dia interactuar con los stake.....

### Métricas de Evaluación:

Answer Relevancy: 0.45 (Threshold: 0.5) - The score is 0.45 because the actual output has several irrelevant statements that deviate from addressing the input question about the components of

software, but still provides some related information.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first node, which is not relevant to software components due to discussing 'aspectos a tratar practicas para el agilismo...', is ranked higher than the second node, which is relevant and defines the components and their relationships. Additionally, the third node, which is not relevant as it discusses the development process, is ranked higher than the second node.

Contextual Recall: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context partially matches the expected output, specifically the 2nd node mentioning 'arquitectura' and 'documentacion' which relates to sentences I and D, but lacks information about other software components.

Contextual Relevancy: 0.47 (Threshold: 0.5) - The score is 0.47 because the retrieval context is mostly irrelevant to the input, with statements like 'arquitectura de software desarrollo de software i' and 'ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios' having no relation to the components of software according to the definition, but a few relevant statements like 'una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla' and 'define componentes y sus relaciones la arquitectura describe las multiples facetas de la estructura del software' brought the score up slightly.

## Caso de Prueba: 34 - Fallo

### Input del Usuario:

¿Cuál es la definición de software en un sistema informático?

### Respuesta Generada:

Según el Documento 1, la definición de software en un sistema informático es: "El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que, al ser ejecutadas por un sistema informático (hardware), le permiten realizar tareas específicas y procesar información." En otras palabras, el software se refiere al conjunto de instrucciones y programas que se ejecutan en un sistema informático, permitiendo al hardware realizar tareas específicas y procesar información de manera adecuada.

### Respuesta Esperada:

Según el contexto, la definición de software en un sistema informático es: "un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un sistema informático (hardware), le permiten realizar tareas específicas y procesar información."

### Contexto Relevante:

1. Conceptos ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un sistema informático (hardware), le permiti.....

### Contexto Recuperado:

1. conceptos que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un sistema informatico hardware le permiten rea.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the response perfectly addressed the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 and node 2, which provide the exact definition of software, are ranked higher than the irrelevant node, like node 3, which talks about a different topic, 'proceso para especificar requerimientos'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly matches the 1st node in the retrieval context, indicating a flawless alignment.

Contextual Relevancy: 0.28 (Threshold: 0.5) - The score is 0.28 because the retrieval context is mostly irrelevant, with statements like 'The statement is about elicitation, not the definition of software.' and 'The statement is about specifying software functionalities, not the definition of software.' dominating the context, while only a few statements like 'el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion...' provide some relevance to the input.

## Caso de Prueba: 35 - Fallo

### Input del Usuario:

¿Cuál es el componente fundamental del que se compone el software según la definición?

### Respuesta Generada:

Según el Documento 3, el software es un conjunto estructurado de "instrucciones codificadas en un lenguaje de programación". Por lo tanto, el componente fundamental del que se compone el software es una instrucción codificada en un lenguaje de programación.

### Respuesta Esperada:

Según la definición, el componente fundamental del que se compone el software es "Instrucciones (programas de computador) que cuando se ejecutan proveen la funcionalidad y rendimiento requerido."

### Contexto Relevante:

1. Conceptos ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un sistema informático (hardware), le permiti.....

### Contexto Recuperado:

1. de software arquitectura a b c una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla asi como los princ.....
2. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....
3. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....

### Métricas de Evaluación:

Answer Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the output provides a definition of software but fails to identify the fundamental component, which is the main focus of the input question.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, and it's ranked third, with the first two nodes being irrelevant, such as node 1 with the reason 'de software arquitectura...' does not mention the fundamental component of software, and node 2 with the reason 'conceptos...' talks about coding, testing, and software development life cycle, but does not address the fundamental component of

software.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output sentence perfectly matches the information provided in the 3rd node in retrieval context, ensuring a flawless recall.

Contextual Relevancy: 0.30 (Threshold: 0.5) - The score is 0.30 because the retrieval context mostly talks about software architecture, development, and engineering, which are irrelevant to the input question about the fundamental component of software, as stated in the reasons for irrelevancy, such as 'The statement is irrelevant because it talks about what the architecture describes, not what the fundamental component of software is.' and 'The retrieval context contained the information about software engineering, which has nothing to do with the fundamental component of software.'



## Caso de Prueba: 36 - Fallo

### Input del Usuario:

¿Cuál es el papel de las estructuras de datos en un software?

### Respuesta Generada:

Según el Documento 1, las estructuras de datos en un software permiten a los programas manipular la información adecuadamente. En otras palabras, las estructuras de datos son fundamentales en un software porque permiten organizar, almacenar y recuperar la información de manera eficiente, lo que a su vez permite que el software realice tareas específicas y procese información. En resumen, el papel de las estructuras de datos en un software es proporcionar una forma de organizar y manipular la información para que el software pueda realizar sus funciones correctamente.

### Respuesta Esperada:

El papel de las estructuras de datos en un software es permitir a los programas manipular la información adecuadamente.

### Contexto Relevante:

1. Conceptos ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un sistema informático (hardware), le permiti.....

### Contexto Recuperado:

1. conceptos que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un sistema informatico hardware le permiten rea.....

2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....

3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the response partially addresses the role of data structures in software, but gets sidetracked by discussing the general purpose and correct functioning of software.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as the second node 'tareas investigar la diferencia entre get y post...' and the third node 'requerimientos no funcionales tecnicas para validacion de requerimientos...', are correctly ranked lower than the first node, which is a direct match to the expected output.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, which mentions the exact same concept of 'estructuras de datos' and their role in software.

Contextual Relevancy: 0.08 (Threshold: 0.5) - The score is 0.08 because the retrieval context mostly contained irrelevant information, such as definitions of software, documentation, HTTP methods, and other unrelated topics, with only one relevant statement about 'estructuras de datos que permiten a los programas manipular la informacion adecuadamente'.

## Caso de Prueba: 37 - Fallo

### Input del Usuario:

¿Cuál es la característica del software que se refiere a su capacidad de ser modificado o adaptado con facilidad?

### Respuesta Generada:

Según el contexto proporcionado, la característica del software que se refiere a su capacidad de ser modificado o adaptado con facilidad es la "mantenibilidad". Esta característica se menciona en el Documento 1, donde se lista como uno de los atributos de calidad del software. La mantenibilidad se define como la facilidad para efectuar cambios en el software.

### Respuesta Esperada:

La característica del software que se refiere a su capacidad de ser modificado o adaptado con facilidad es la maleabilidad.

### Contexto Relevante:

1. Conceptos 02 04 06 El software no es usado como un automovil pero se deteriora C a r a c t e r í s t i c a s d e l S o f t w a r e El software puede depender de la plataforma Los problemas del software s.....

### Contexto Recuperado:

1. t e s t e a b i l i d a d se debe disenar sus componentes para que sean facilmente probados d e s p l e g a b i l i d a d facilidad con la que se puede pasar de desarrollo a produccion e v o l u c i o n r e l e a s.....
2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. de software arquitectura a b c una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla asi como los princ.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the response is mostly relevant, but mentions a document which is not directly related to the software characteristic, preventing a perfect score.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node, which mentions 'm a n t e n i b i l i d a d facilidad para efectuarle cambios', directly relating to the input, ranked higher than the irrelevant nodes, such as the second node discussing 'requerimientos del software' and the third node describing 'arquitectura de software', which do not address the input's concept.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the concept of maleabilidad.

Contextual Relevancy: 0.12 (Threshold: 0.5) - The score is 0.12 because the retrieval context is mostly irrelevant, with statements talking about testability, deployability, evolvability, and other unrelated topics, as stated in the reasons for irrelevancy, and only a few statements like 'm a n t e n i b i l i d a d facilidad para efectuarle cambios' and 'una característica o restricción operativa funcional o de diseño de un producto o proceso' are actually relevant to the input.

## Caso de Prueba: 38 - Fallo

### Input del Usuario:

¿Cuál es la característica del software que hace que su construcción sea intensiva en términos humanos?

### Respuesta Generada:

Según el documento 5, la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. Esto sugiere que la característica del software que hace que su construcción sea intensiva en términos humanos es la necesidad de alta calidad y satisfacción de las necesidades y expectativas de los usuarios. Además, según el documento 3, la arquitectura del software describe las múltiples facetas de la estructura del software, el comportamiento, el diseño, la codificación y la documentación. Esto sugiere que la construcción de software implica una gran cantidad de esfuerzo humano para diseñar, codificar, probar y documentar el software de manera que cumpla con los requisitos y expectativas de los usuarios. En resumen, la característica del software que hace que su construcción sea intensiva en términos humanos es la necesidad de alta calidad, satisfacción de las necesidades y expectativas de los usuarios, y la complejidad de la estructura del software que requiere un gran esfuerzo humano para su diseño, codificación, pruebas y documentación.

### Respuesta Esperada:

La característica del software que hace que su construcción sea intensiva en términos humanos es que es "humanintensive", es decir, requiere una gran cantidad de esfuerzo y trabajo humano.

### Contexto Relevante:

1. Conceptos 02 04 06 El software no es usado como un automovil pero se deteriora C a r a c t e r í s t i c a s d e l S o f t w a r e El software puede depender de la plataforma Los problemas del software s.....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

2. quien es el dueno del web no existe un individuo o entidad sola que sea dueno del web es un consorcio internacional de caracter abierto y libre pero existen ciertas entidades a nivel mundial que lider.....

3. de software arquitectura a b c una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla asi como los princ.....

### Métricas de Evaluación:

Answer Relevancy: 0.41 (Threshold: 0.5) - The score is 0.41 because the actual output is mostly irrelevant, discussing various aspects of software engineering and construction phases, but barely touching on the characteristic that makes software construction intensive in human terms.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 with reason ' "quien es el dueño del web..." is not related to the topic of software construction being human-intensive.' and node 3 with reason ' "de software arquitectura..." discusses software architecture, which is not directly related to the human-intensive aspect of software construction.', are correctly ranked lower than the relevant node 1 with reason ' "aplicación de principios de ingeniería la ingeniería de software utiliza conceptos de ingeniería como el análisis de requisitos el diseño la construcción las pruebas y el mantenimiento para garantizar que el software sea confiable eficiente y escalable" implies that software construction requires a great deal of human effort and work.'

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions 'ingeniería de software', directly relating to the concept of 'human-intensive' software construction, resulting in a flawless match.

Contextual Relevancy: 0.14 (Threshold: 0.5) - The score is 0.14 because the retrieval context mostly talks about unrelated topics such as project management, software development life cycle, software quality, methodologies, and teamwork, as stated in the reasons for irrelevancy, and only briefly mentions software engineering principles and W3C standards development, which are somewhat relevant to the input.

## Caso de Prueba: 39 - Fallo

### Input del Usuario:

¿Cuál es la característica del software que hace que su construcción sea intensiva en mano de obra, pero se esté volviendo cada vez más automatizada?

### Respuesta Generada:

La característica del software que hace que su construcción sea intensiva en mano de obra, pero se esté volviendo cada vez más automatizada, es la calidad. Según el Documento 1, la calidad del software se centra en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable. La construcción de software de alta calidad requiere la aplicación de conceptos de ingeniería, como el análisis de requisitos, el diseño, la construcción, las pruebas y el mantenimiento, lo que implica un proceso intensivo en mano de obra. Sin embargo, según el Documento 10, los procesos de desarrollo ágil intentan que el despliegue sea cada vez más automático, mediante la utilización de infraestructura como código (IaC) y métodos de despliegue continuo, lo que indica que la construcción de software se está volviendo cada vez más automatizada.

### Respuesta Esperada:

La característica del software que hace que su construcción sea intensiva en mano de obra, pero se esté volviendo cada vez más automatizada, es que es "maleable".

### Contexto Relevante:

1. Conceptos 02 04 06 El software no es usado como un automovil pero se deteriora C a r a c t e r í s t i c a s d e l S o f t w a r e El software puede depender de la plataforma Los problemas del software s.....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

2. quien es el dueno del web no existe un individuo o entidad sola que sea dueno del web es un consorcio internacional de caracter abierto y libre pero existen ciertas entidades a nivel mundial que lider.....

3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

### Métricas de Evaluación:

Answer Relevancy: 0.43 (Threshold: 0.5) - The score is 0.43 because the output touches on related concepts, but fails to directly address the question, leaving some relevant information unexplained.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'construccion' and 'desarrollo de software', is relevant to the input, and the third-ranked node, which mentions 'codificacion' and 'ciclo de vida del desarrollo de software', is also relevant, but the second-ranked node, which is about 'quien es el dueño del web', is an irrelevant node that should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, as the sentence about software being 'maleable' is directly attributed to the 1st node in the retrieval context, which discusses key concepts of software engineering.

Contextual Relevancy: 0.20 (Threshold: 0.5) - The score is 0.20 because the retrieval context is mostly irrelevant, with statements like 'The statement talks about the ownership and development of the web...' and 'The retrieval context contained the information 'las pruebas de software...'' having no relation to the input, and only a few relevant statements like 'el ciclo de vida del desarrollo de software...' and 'la ingeniería de software utiliza conceptos de ingeniería...' being present.



## Caso de Prueba: 40 - Fallo

### Input del Usuario:

¿Cuál es la característica del software que hace que no se pueda tocar ni ver, pero sí utilizar?

### Respuesta Generada:

Según el Documento 2, la característica del software que hace que no se pueda tocar ni ver, pero sí utilizar, es que el software es **intangible**. Esto significa que el software no tiene una forma física tangible, pero sigue siendo posible utilizarlo y interactuar con él.

### Respuesta Esperada:

La característica del software que hace que no se pueda tocar ni ver, pero sí utilizar, es que es intangible.

### Contexto Relevante:

1. Conceptos 02 04 06 El software no es usado como un automovil pero se deteriora C a r a c t e r í s t i c a s d e l S o f t w a r e El software puede depender de la plataforma Los problemas del software s.....

### Contexto Recuperado:

1. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....
2. conceptos el software no es usado como un automovil pero se deteriora c a r a c t e r í s t i c a s d e l s o f t w a r e el software puede depender de la plataforma los problemas del software son usualme.....
3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the response partially addresses the input question, but lacks clear explanations for why the software cannot be touched or seen, focusing only on its usability.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first and third nodes in the retrieval contexts, which are irrelevant to the input, were ranked higher than the second node, which directly answers the question about the characteristic of software, stating 'el software es intangible y difícil de medir'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.20 (Threshold: 0.5) - The score is 0.20 because the retrieval context is mostly irrelevant, with statements like 'conceptos la codificacion...', 'las pruebas de software...', and 'el ciclo de vida del desarrollo de software...' that do not relate to the characteristic of software being intangible, but only a few relevant statements like 'el software es intangible' and 'caracteristicas del software' that match the input.

## Caso de Prueba: 41 - Fallo

### Input del Usuario:

¿Cuál es la característica del software que lo hace complejo de mantener y desarrollar?

### Respuesta Generada:

Según el Documento 5, la característica del software que lo hace complejo de mantener y desarrollar es su "infinito crecimiento" y la necesidad de "reutilización para reducir costos", lo que requiere "mantenimiento" y la adopción de "nuevas tecnologías para evitar obsolescencia". Esto sugiere que la complejidad del software se debe a su capacidad de crecer y evolucionar con el tiempo, lo que requiere una mayor complejidad en su diseño y desarrollo.

### Respuesta Esperada:

La característica del software que lo hace complejo de mantener y desarrollar es que los problemas del software son usualmente complejos.

### Contexto Relevante:

1. Conceptos 02 04 06 El software no es usado como un automovil pero se deteriora C a r a c t e r í s t i c a s d e l S o f t w a r e El software puede depender de la plataforma Los problemas del software s.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....
2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant, but slightly derailed by mentioning maintenance as a consequence rather than a characteristic, which prevented a perfect score.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the 1st node which talks about agile software development, the 2nd node which explains software requirements, and the 3rd node which is a duplicate of the 2nd node, are irrelevant to the input and should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the sentence perfectly aligns with the 1st node in the retrieval context, capturing the essence of 'desarrollo de software agil' with its mention of 'software' and 'desarrollar'.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context mostly talks about agile software development, alternative practices, and project management, which are not related to the complexity of maintaining and developing software, as stated in the reasons for irrelevancy, whereas only a few statements about requirements and their specifications are relevant to the input.

## Caso de Prueba: 42 - Fallo

### Input del Usuario:

¿Cuál es el enfoque principal de la ingeniería de software en la creación de software?

### Respuesta Generada:

Según el contexto proporcionado, el enfoque principal de la ingeniería de software en la creación de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. (Documento 1 y Documento 2). La ingeniería de software se enfoca en aplicar principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas y mantenimiento de software, con el objetivo de crear software de alta calidad que cumpla con los requisitos y expectativas de los usuarios.

### Respuesta Esperada:

El enfoque principal de la ingeniería de software en la creación de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Contexto Relevante:

1. Conceptos Que es Ingeniería de Software? La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, de.....

### Contexto Recuperado:

1. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....

3. presentacion de curso características del curso definiciones que es software que es ingenieria de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....

### Métricas de Evaluación:

Answer Relevancy: 0.30 (Threshold: 0.5) - The score is 0.30 because the actual output contains some relevant aspects of software engineering, but they are not the main focus and are overshadowed by irrelevant statements.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts directly addresses the question, and the third node is also relevant, but the second node, ranked 2, is not directly related to the main focus of software engineering in creating software,

which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly matches the 1st node in the retrieval context, demonstrating a seamless alignment between the two.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context has many irrelevant statements that are too broad or not specifically related to the main focus of software engineering in creating software, such as 'conceptos que es ingenieria de software', 'que es software', and 'ciclo de vida', whereas only a few statements like 'la ingenieria de software se enfoca en la creacion sistematica y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios' are actually relevant.

## Caso de Prueba: 43 - Éxito

### Input del Usuario:

¿Cuáles son los principios y prácticas clave que se aplican en la ingeniería de software para crear sistemas de alta calidad?

### Respuesta Generada:

Según los documentos proporcionados, los principios y prácticas clave que se aplican en la ingeniería de software para crear sistemas de alta calidad son: 1. **\*\*Aplicación de principios de ingeniería\*\***: La ingeniería de software utiliza conceptos de ingeniería como el análisis de requisitos, el diseño, la construcción, las pruebas y el mantenimiento para garantizar que el software sea confiable, eficiente y escalable. 2. **\*\*Gestión de proyectos\*\***: Aplicar técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que se entregue a tiempo y dentro del presupuesto. 3. **\*\*Ciclo de vida del desarrollo de software\*\***: Seguir un proceso estructurado para guiar el desarrollo del software desde la concepción hasta la implementación y el mantenimiento, incluyendo fases como planificación, análisis, diseño, codificación, pruebas y despliegue. 4. **\*\*Calidad del software\*\***: Centrarse en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable. 5. **\*\*Metodologías y prácticas\*\***: Emplear diversas metodologías (como Agile, Scrum, Cascada, etc.) y prácticas (como control de versiones, integración continua, etc.) para gestionar el desarrollo de software de manera efectiva. 6. **\*\*Trabajo en equipo\*\***: Fomentar la colaboración entre equipos multidisciplinarios, incluyendo desarrolladores, analistas, diseñadores, testers y otros profesionales para lograr los objetivos del proyecto. Estos principios y prácticas clave permiten a los ingenieros de software crear sistemas de alta calidad que satisfagan las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

Según el contexto, los principios y prácticas clave que se aplican en la ingeniería de software para crear sistemas de alta calidad son la aplicación de principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, despliegue, monitoreo y mantenimiento de software, con el fin de crear software de alta calidad de manera sistemática y eficiente.

### Contexto Relevante:

1. Conceptos Que es Ingeniería de Software? La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, de.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

3. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a comprehensive and accurate response without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts directly answers the question about principles and practices in software engineering, and the third-ranked node lists key aspects of software engineering relevant to creating high-quality systems, but the second-ranked node, which only defines software and repeats the definition of software engineering, should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mirrors the sentence structure and content, resulting in a seamless match.

Contextual Relevancy: 0.91 (Threshold: 0.5) - The score is 0.91 because the retrieval context is highly relevant to the input, as it directly addresses the principles and practices of software engineering, such as 'la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad' and 'aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue monitoreo y mantenimiento de software', which aligns with the input's question about creating high-quality systems.



## Caso de Prueba: 44 - Fallo

### Input del Usuario:

¿Cuál es el enfoque principal de la ingeniería de software al crear software de alta calidad?

### Respuesta Generada:

Según el Documento 1, el enfoque principal de la ingeniería de software al crear software de alta calidad es "la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios". Esto significa que la ingeniería de software se enfoca en aplicar principios, metodologías y prácticas de la ingeniería para crear software que cumpla con los requisitos y necesidades de los usuarios de manera sistemática y eficiente.

### Respuesta Esperada:

Según el contexto, el enfoque principal de la ingeniería de software al crear software de alta calidad es la creación sistemática y eficiente de software que satisfaga las necesidades y expectativas de los usuarios.

### Contexto Relevante:

1. Conceptos Que es Ingeniería de Software? La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, de.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

3. aplicar conocimiento la ingeniería de software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad a l t a d e m.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as node 1 which clearly addresses the question, are ranked higher than the irrelevant nodes, like node 2 which only talks about software and its definition, and node 3 which discusses the importance of software engineering but doesn't directly address the main focus.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless alignment between the two.

Contextual Relevancy: 0.29 (Threshold: 0.5) - The score is 0.29 because the retrieval context mostly contained irrelevant information, such as definitions of software engineering and software, and only a few statements like 'la ingenieria de software se enfoca en la creacion sistematica y eficiente de software de alta calidad' were directly related to the main focus of software engineering in creating high-quality software.

## Caso de Prueba: 45 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de las pruebas de software en el ciclo de vida de desarrollo de un sistema de software?

### Respuesta Generada:

Según el Documento 1, el propósito principal de las pruebas de software es "evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer". En otras palabras, las pruebas de software tienen como objetivo principal asegurarse de que el sistema o aplicación de software cumpla con los requisitos y expectativas establecidos, detectando y corrigiendo posibles errores o defectos en el proceso.

### Respuesta Esperada:

No se puede determinar el propósito principal de las pruebas de software en el ciclo de vida de desarrollo de un sistema de software con el contexto proporcionado, ya que no se menciona explícitamente el propósito de las pruebas de software. El contexto solo menciona que la ingeniería de software incluye pruebas, pero no proporciona información adicional sobre su objetivo.

### Contexto Relevante:

1. Conceptos Que es Ingeniería de Software? La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, de.....

### Contexto Recuperado:

1. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stakeholders.....

3. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de prácticas principios y procedimientos que guían y regulan el proceso de creación.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the output is mostly relevant, but it strays from the main purpose of software testing by focusing on correcting errors, which is not the primary focus.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the 1st node stating 'pruebas de software es el proceso de evaluar y verificar que un sistema o aplicacion de software hace lo que se supone que debe hacer', the 2nd node mentioning 'evaluar la calidad del sistema para encontrar y corregir posibles errores', and the 3rd node discussing 'metodologias de desarrollo de software', are irrelevant to the input and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.15 (Threshold: 0.5) - The score is 0.15 because the retrieval context is largely irrelevant, with most statements discussing broader topics like the software development life cycle, stakeholder interaction, and system deployment, as mentioned in the reasons for irrelevancy, whereas only a few statements like 'las pruebas de software es el proceso de evaluar y verificar que un sistema o aplicacion de software hace lo que se supone que debe hacer' and 'evaluar la calidad del sistema para encontrar y corregir posibles errores pruebas' are actually relevant to the main purpose of software testing.

## Caso de Prueba: 46 - Fallo

### Input del Usuario:

¿Cuál es el papel del mantenimiento en la ingeniería de software?

### Respuesta Generada:

Según el contexto proporcionado, el papel del mantenimiento en la ingeniería de software es uno de los aspectos clave de la disciplina. EnSpecificamente, en el Documento 1 y 2 se menciona que la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios, y que incluye el desarrollo, implementación, pruebas, despliegue, monitoreo y mantenimiento del software. En el Documento 3, se destaca que el mantenimiento es una de las fases del ciclo de vida del desarrollo de software, junto con la planificación, análisis, diseño, codificación, pruebas y despliegue. El mantenimiento es esencial para garantizar que el software sea confiable, eficiente y escalable. En resumen, el papel del mantenimiento en la ingeniería de software es garantizar que el software siga funcionando correctamente y satisfaciendo las necesidades de los usuarios una vez que ha sido desarrollado y desplegado. Esto implica realizar actualizaciones, correcciones y mejoras al software para asegurar su calidad yperformance en el tiempo.

### Respuesta Esperada:

Según el contexto, el papel del mantenimiento en la ingeniería de software es una de las etapas del proceso de creación de software, junto con el diseño, desarrollo, implementación, pruebas, despliegue y monitoreo.

### Contexto Relevante:

1. Conceptos Que es Ingeniería de Software? La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, de.....

### Contexto Recuperado:

1. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....

3. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the response partially addresses the role of maintenance in software engineering, but includes some irrelevant information about the focus and activities of software development.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'el mantenimiento de software' as a stage in the software creation process, is correctly ranked high, and the third-ranked node, which mentions 'mantenimiento' as a phase in the software development life cycle, is also correctly ranked high. However, the second-ranked node, which is not directly related to the role of maintenance in software engineering, should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions all the stages of software creation, resulting in a flawless match.

Contextual Relevancy: 0.46 (Threshold: 0.5) - The score is 0.46 because the retrieval context is mostly irrelevant, with statements like 'conceptos que es ingenieria de software' and 'gestion de proyectos' that have nothing to do with the role of maintenance in software engineering, but it does contain some relevant information like 'el mantenimiento de software' and 'la ingenieria de software utiliza conceptos de ingenieria como el analisis de requisitos el diseno la construccion las pruebas y el mantenimiento'.

## Caso de Prueba: 47 - Fallo

### Input del Usuario:

¿Cuál es el papel de la gestión de proyectos en la Ingeniería de software?

### Respuesta Generada:

Según el Documento 1, la gestión de proyectos en la Ingeniería de software se centra en aplicar técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que se entregue a tiempo y dentro del presupuesto. En otras palabras, la gestión de proyectos en la Ingeniería de software tiene como objetivo guiar el desarrollo del software desde su concepción hasta su implementación y mantenimiento, garantizando que el software se entregue dentro de los plazos y recursos establecidos. En este sentido, la gestión de proyectos en la Ingeniería de software es crucial para asegurar que el software se desarrolla de manera eficiente, confiable y escalable, y que se satisfagan las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

El papel de la gestión de proyectos en la Ingeniería de software es aplicar técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que se entregue a tiempo y dentro del presupuesto.

### Contexto Relevante:

1. Conceptos: Aspectos clave de la Ingeniería de software 5. Gestión de proyectos Aplica técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que .....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....
2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....
3. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant answer to the question about the role of project management in software engineering!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly answering the question about the role of project management in software engineering, and the irrelevant nodes, such as node 2 discussing the software development life cycle and node 3 providing a general definition of software engineering, are correctly ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in retrieval context, which mirrors the same project management techniques for software development.

Contextual Relevancy: 0.45 (Threshold: 0.5) - The score is 0.45 because the retrieval context has many irrelevant statements that describe software development life cycle, software quality, teamwork, and what software is, which are not directly related to the role of project management in software engineering, and only a few relevant statements that mention project management, such as 'gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software'.



## Caso de Prueba: 48 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal del ciclo de vida del desarrollo de software en la ingeniería de software?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal del ciclo de vida del desarrollo de software en la ingeniería de software es guiar el desarrollo del software desde la concepción hasta la implementación y el mantenimiento, mediante un proceso estructurado que incluye fases como planificación, análisis, diseño, codificación, pruebas y despliegue. Este ciclo de vida permite entender los requisitos de los usuarios, diseñar y construir un sistema de software que se adapte a esas necesidades, y asegurar que el software sea confiable, eficiente, escalable y de alta calidad. En otras palabras, el ciclo de vida del desarrollo de software es un marco para organizar y gestionar las tareas involucradas en cada fase del desarrollo de software, con el fin de mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo.

### Respuesta Esperada:

El propósito principal del ciclo de vida del desarrollo de software en la ingeniería de software es guiar el desarrollo del software desde la concepción hasta la implementación y el mantenimiento, asegurando que se cumplan los requisitos y se entregue un software de calidad.

### Contexto Relevante:

1. Conceptos: Aspectos clave de la Ingeniería de software 5. Gestión de proyectos Aplica técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que .....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stake.....

3. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### Métricas de Evaluación:

Answer Relevancy: 0.88 (Threshold: 0.5) - The score is 0.88 because the response accurately addresses the primary purpose of the software development life cycle, but briefly mentions a related yet non-essential aspect, minimizing risks.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, directly address the question and clearly explain the purpose of the software development life cycle, making them all highly relevant.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, ensuring a flawless match.

Contextual Relevancy: 0.58 (Threshold: 0.5) - The score is 0.58 because the retrieval context contains many irrelevant statements, such as 'presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software', which have nothing to do with the main purpose of the software development life cycle, but also includes some relevant statements like 'ciclo de vida proceso de creacion de software proceso o marco estructurado para el desarrollo de software', which are directly related to the input.

## Caso de Prueba: 49 - Fallo

### Input del Usuario:

¿Cuál es el aspecto clave de la Ingeniería de software que se enfoca en la identificación y documentación de las necesidades y expectativas del cliente para desarrollar un software que se ajuste a sus requerimientos?

### Respuesta Generada:

Según el contexto proporcionado, el aspecto clave de la Ingeniería de software que se enfoca en la identificación y documentación de las necesidades y expectativas del cliente para desarrollar un software que se ajuste a sus requerimientos es la "Gestión de Proyectos" ( mencionada en el Documento 1) y más específicamente, el proceso de "Elicitación y Análisis de Requerimientos" (mencionado en el Documento 5). La Elicitación de Requerimientos es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. El Análisis de Requerimientos es el proceso de identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. Ambos procesos son fundamentales para desarrollar un software que se ajuste a los requerimientos del cliente, ya que permiten identificar y documentar las necesidades y expectativas del cliente, lo que a su vez guiará el diseño, desarrollo y pruebas del software.

### Respuesta Esperada:

El aspecto clave de la Ingeniería de software que se enfoca en la identificación y documentación de las necesidades y expectativas del cliente para desarrollar un software que se ajuste a sus requerimientos es la "Aplicación de principios de ingeniería", específicamente el análisis de requisitos.

### Contexto Relevante:

1. Conceptos: Aspectos clave de la Ingeniería de software 5. Gestión de proyectos Aplica técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que .....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....
2. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....
3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a lta d e m.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the output correctly addressed the key aspect of software engineering, but was partially derailed by incorrect and off-topic statements about project management.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which only talks about 'conceptos que es ingeniería de software', and node 3 which focuses on 'aplicar conocimiento' to create quality software, are correctly ranked lower than the highly relevant node 1 that specifically addresses 'análisis de requisitos' for identifying and documenting client needs.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, specifically the 1st node, which mentions 'aplicación de principios de ingeniería' and 'análisis de requisitos', accurately capturing the essence of software engineering.

Contextual Relevancy: 0.23 (Threshold: 0.5) - The score is 0.23 because the retrieval context mostly discusses unrelated topics such as project management, software development life cycle, software quality, and teamwork, as stated in the reasons for irrelevancy, whereas only a few statements like 'la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios' are actually relevant to the input.

## Caso de Prueba: 50 - Fallo

### Input del Usuario:

¿Cuál es el aspecto clave de la ingeniería de software que se centra en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable?

### Respuesta Generada:

Según el Documento 1 y Documento 2, el aspecto clave de la ingeniería de software que se centra en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable es la "Calidad del software". Esta se enfoca en asegurar que el software cumpla con los objetivos de ser funcional, confiable, usable, eficiente, mantenible y portable.

### Respuesta Esperada:

El aspecto clave de la ingeniería de software que se centra en la creación de software que sea funcional, confiable, usable, eficiente, mantenible y portable es la "Calidad del software".

### Contexto Relevante:

1. Conceptos: Aspectos clave de la Ingeniería de software 5. Gestión de proyectos Aplica técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que .....

### Contexto Recuperado:

1. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....
2. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....
3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l ta d e m.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 and node 2, are ranked higher than irrelevant nodes, as they directly answer the question by mentioning '4 calidad del software se centra en la creacion de software que sea funcional confiable usable eficiente mantenible y portable', whereas node 3 is correctly ranked lower due to not mentioning the key aspect of software engineering.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 4th node in retrieval context, which mirrors the exact same software quality aspects.

Contextual Relevancy: 0.22 (Threshold: 0.5) - The score is 0.22 because the retrieval context is mostly irrelevant to the input, with reasons stating that the context is about project management, software development life cycle, teamwork, and other aspects not directly related to creating functional, reliable, usable, efficient, maintainable, and portable software, as quoted in the reasons for irrelevancy.

## Caso de Prueba: 51 - Éxito

### Input del Usuario:

¿Cuál es la metodología ágil que se enfoca en la entrega rápida y flexible de software, y se basa en la colaboración y comunicación entre el equipo de desarrollo y los stakeholders?

### Respuesta Generada:

Basándome en los documentos proporcionados, puedo concluir que la metodología ágil que se enfoca en la entrega rápida y flexible de software, y se basa en la colaboración y comunicación entre el equipo de desarrollo y los stakeholders, es Scrum. En el Documento 3, se menciona que en las metodologías ágiles, un Release Plan (Plan de Entregas) es un documento o estrategia que define cómo y cuándo se entregarán las funcionalidades de un producto a los usuarios finales o al cliente. Esto indica que Scrum se enfoca en la entrega incremental de software, lo que se alinea con la idea de entrega rápida y flexible. Además, en el Documento 4, se menciona que Scrum implica un incremento de software a través de pequeñas entregas, lo que sugiere que esta metodología se enfoca en la entrega rápida y flexible de software. En resumen, Scrum es la metodología ágil que se enfoca en la entrega rápida y flexible de software, y se basa en la colaboración y comunicación entre el equipo de desarrollo y los stakeholders.

### Respuesta Esperada:

La metodología ágil que se enfoca en la entrega rápida y flexible de software, y se basa en la colaboración y comunicación entre el equipo de desarrollo y los stakeholders es una de las metodologías mencionadas en el punto 3, pero no se especifica cuál es exactamente. Sin embargo, se puede inferir que se refiere a una metodología ágil, ya que se menciona la entrega rápida y flexible, que es un aspecto clave de las metodologías ágiles.

### Contexto Relevante:

1. Conceptos: Aspectos clave de la Ingeniería de software 5. Gestión de proyectos Aplica técnicas de gestión de proyectos para planificar, organizar y controlar el desarrollo de software, asegurando que .....

### Contexto Recuperado:

1. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. practicas agiles en scrum release plan plan de entregas en las metodologias agiles un release plan plan de entregas es un documento o estrategia que define como y cuando se entregaran las funcionalida.....

### Métricas de Evaluación:

Answer Relevancy: 0.88 (Threshold: 0.5) - The score is 0.88 because the response is mostly relevant, but slightly derailed by a tangential mention of Release Plan, making it not a perfect fit.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1 to 3, directly address the question's key aspects of agile methodologies, such as 'entrega incremental', 'colaboracion en equipo', and 'plan de entregas en las metodologias agiles', making them perfectly relevant and correctly ranked.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the nodes in retrieval context, with multiple sentences matching key aspects of agile methodology mentioned in nodes 1 and 3.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context is only partially relevant, as it discusses agile methodologies in general, but lacks specific information about the methodology that focuses on rapid and flexible software delivery, as asked in the input.



## Caso de Prueba: 52 - Fallo

### Input del Usuario:

¿Cuál es la relación entre la entrega tardía y el sobrecosto en proyectos de software, y cómo afectan a la calidad del producto final?

### Respuesta Generada:

Según el Documento 1, la entrega tardía y el sobrecosto son problemas comunes en proyectos de software que afectan negativamente la calidad del producto final. La entrega tardía se refiere a la incapacidad de entregar el software dentro del plazo estimado, lo que puede provocar retrasos en la implementación y el lanzamiento del producto. Por otro lado, el sobrecosto se refiere a la superación del presupuesto inicial, lo que puede generar problemas financieros y de recursos. La relación entre la entrega tardía y el sobrecosto es directa. Cuando un proyecto de software se retrasa, es probable que se necesiten más recursos y tiempo para cumplir con los requisitos del proyecto, lo que conduce a un aumento en los costos. A su vez, el sobrecosto puede generar presión adicional en el equipo de desarrollo para entregar el producto lo antes posible, lo que puede afectar la calidad del producto final. En proyectos de software, la calidad del producto final se ve afectada negativamente cuando se entregan productos con defectos, bugs o funcionalidades incompletas. Esto se debe a que se han sacrificado la calidad y la prueba exhaustiva del producto en favor de una entrega rápida y barata. Sin embargo, esta estrategia a corto plazo puede generar costos adicionales a largo plazo, ya que se necesitarán más recursos para corregir los errores y defectos del producto. Para evitar estos problemas, las metodologías ágiles, como se menciona en el Documento 2, enfatizan la entrega temprana y continua de software con valor, con el fin de satisfacer las necesidades del cliente y entregar productos de alta calidad. Además, la planificación y la estimación adecuadas de los recursos y tiempos, como se menciona en el Documento 6, son clave para evitar la entrega tardía y el sobrecosto en proyectos de software. En resumen, la entrega tardía y el sobrecosto en proyectos de software están estrechamente relacionados y pueden afectar negativamente la calidad del producto final. Para evitar estos problemas, es importante implementar prácticas y metodologías ágiles que enfatizan la entrega temprana y continua de software con valor, así como una planificación y estimación adecuadas de los recursos y tiempos.

### Respuesta Esperada:

La relación entre la entrega tardía y el sobrecosto en proyectos de software es directa. Cuando un proyecto de software se entrega tarde, es probable que supere el presupuesto estimado, lo que se conoce como sobrecosto. Esto se debe a que se requiere más tiempo y recursos para completar el proyecto, lo que aumenta los costos. A su vez, la entrega tardía y el sobrecosto pueden afectar negativamente la calidad del producto final, ya que se pueden tomar atajos o sacrificar características para cumplir con el plazo de entrega, lo que puede llevar a un producto de baja calidad.

### Contexto Relevante:

1. Conceptos Principales problema de los proyectos de software Problemas de calidad Entregas tardías Sobrecosto (fuera del presupuesto estimado)...

### Contexto Recuperado:

1. conceptos principales problema de los proyectos de software problemas de calidad entregas tardias sobrecosto fuera del presupuesto estimado...
2. nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor metodologias agiles 12 principios definidos en el manifiesto agil aceptamos que los requi.....
3. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node directly addressing the input question by mentioning key concepts related to the problem of software projects, including late deliveries and cost overruns, while the second and third nodes, ranked lower, do not provide any relevant information about the relationship between late delivery and cost overruns.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to nodes in the retrieval context, specifically the 1st node, which provides strong evidence for the relationship between late delivery and cost overrun in software projects.

Contextual Relevancy: 0.31 (Threshold: 0.5) - The score is 0.31 because the retrieval context mostly lacks relevant information about the relationship between late delivery and cost overrun in software projects, as stated in the reasons for irrelevancy, and only briefly touches on related concepts like 'entregas tardias', 'sobrecosto fuera del presupuesto estimado', and 'problemas de calidad'.

## Caso de Prueba: 53 - Fallo

### Input del Usuario:

¿Cuáles son las principales causas de las entregas tardías en los proyectos de software y cómo afectan al presupuesto y a la satisfacción del cliente?

### Respuesta Generada:

Según el Documento 1, los problemas de calidad, sobrecosto y entregas tardías son algunos de los principales problemas que enfrentan los proyectos de software. Las entregas tardías en los proyectos de software se pueden deber a varias causas, algunas de las cuales pueden ser: \* Falta de planificación y estimación realista del tiempo y recursos necesarios para completar el proyecto. \* Cambios en los requisitos del proyecto sin una adecuada gestión de cambios. \* Complejidad técnica no anticipada en el proyecto. \* Recursos insuficientes o no adecuados para el proyecto. \* Falta de comunicación efectiva dentro del equipo de desarrollo y con el cliente. Estas causas pueden afectar negativamente al presupuesto y a la satisfacción del cliente de varias maneras: \* Sobrecosto: Las entregas tardías pueden requerir recursos adicionales y horas extras para completar el proyecto, lo que puede llevar a un aumento en el costo total del proyecto y sobrepasar el presupuesto inicial. \* Pérdida de confianza: Las entregas tardías pueden afectar la confianza del cliente en el equipo de desarrollo y en la capacidad de entrega de proyectos a tiempo. \* Diferimiento del valor: Las entregas tardías pueden diferir el valor que se entrega al cliente, lo que puede afectar la satisfacción del cliente y la percepción del proyecto. Es importante que los equipos de desarrollo de software tengan en cuenta estas causas y trabajen en estrategias para mitigarlas, como la planificación realista, la gestión de cambios efectiva, la priorización de las características del proyecto y la comunicación efectiva con el cliente.

### Respuesta Esperada:

Las principales causas de las entregas tardías en los proyectos de software son la mala planificación, la falta de recursos, la complejidad del proyecto, la mala gestión del tiempo y los cambios en los requisitos del proyecto. Estas causas pueden generar sobrecostos y afectar negativamente la satisfacción del cliente.

### Contexto Relevante:

1. Conceptos Principales problema de los proyectos de software Problemas de calidad Entregas tardías Sobrecosto (fuera del presupuesto estimado)...

### Contexto Recuperado:

1. conceptos principales problema de los proyectos de software problemas de calidad entregas tardías sobrecosto fuera del presupuesto estimado...
2. navegador web servidor web script script script scripts que corren del lado del servidor navegador web navegador web clientes y servidores web navegador web tarea cuales son las principales diferencia.....
3. tareas investigar que son variables de sesion en una aplicacion web las variables de sesion en una aplicacion web son una forma de almacenar informacion del usuario durante la interaccion con la

aplic.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and comprehensive answer without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 with 'navegador web servidor web script...' and node 3 with 'variables de sesion en una aplicacion web...', are correctly ranked lower than the relevant node 1, which directly addresses the topic of delayed deliveries in software projects.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context only partially supports the expected output, specifically the 1st node in the retrieval context relates to the causes of late deliveries, but lacks information about key concepts like 'satisfacción del cliente' and 'sobrecostos'.

Contextual Relevancy: 0.13 (Threshold: 0.5) - The score is 0.13 because most of the retrieval context statements were deemed irrelevant, with reasons such as 'the statement is not related to the input about causes of late deliveries in software projects' and 'the statement is too general and does not specifically relate to the causes of late deliveries in software projects', with only a few relevant statements like 'entregas tardias' and 'sobrecosto fuera del presupuesto estimado' that barely made a connection.

## Caso de Prueba: 54 - Fallo

### Input del Usuario:

¿Cuáles son las causas más comunes del sobrecosto en proyectos de software?

### Respuesta Generada:

Según el Documento 2, algunos de los problemas comunes en proyectos de software son "sobrecosto fuera del presupuesto estimado". En el Documento 1, se mencionan los costos directos y costos indirectos que se requieren para producir, adquirir, mantener o realizar una actividad, servicio o producto de software. Algunas de las causas más comunes del sobrecosto en proyectos de software pueden incluir: \* Errores en la estimación de tiempos y costos \* Costos directos no previstos, como salarios y beneficios del personal, contratos de consultores, herramientas y software para el desarrollo, infraestructura y equipos, capacitación y formación, gastos de viaje, pruebas y control de calidad, despliegue e implementación en infraestructuras, etc. \* Costos indirectos no previstos, como administración general y gestión, personal administrativo, infraestructura y servicios generales, marketing y ventas, seguros, impuestos, etc. \* Cambios en los requisitos del proyecto que requieren más recursos o tiempo del esperado \* Problemas de calidad que requieren solucionar, lo que puede llevar a un aumento en los costos y tiempo del proyecto \* Mala planificación y gestión del proyecto, lo que puede llevar a una mala utilización de los recursos y a un incremento en los costos. Es importante destacar que la estimación de tiempos y costos es una aproximación informada basada en datos disponibles, experiencia pasada y supuestos razonables, por lo que es importante ser precavido y considerar diferentes escenarios para evitar sobre costos.

### Respuesta Esperada:

Según el contexto, no se proporcionan causas específicas del sobrecosto en proyectos de software. Sin embargo, se menciona que el sobrecosto es uno de los problemas principales de los proyectos de software, junto con problemas de calidad y entregas tardías.

### Contexto Relevante:

1. Conceptos Principales problema de los proyectos de software Problemas de calidad Entregas tardías Sobrecosto (fuera del presupuesto estimado)...

### Contexto Recuperado:

1. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....
2. conceptos principales problema de los proyectos de software problemas de calidad entregas tardias sobrecosto fuera del presupuesto estimado...
3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the response provides some relevant information, but lacks specific details on common causes of cost overruns in software projects, making it only partially relevant to the input question.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first and third nodes in the retrieval contexts, which discuss estimation and costs in agile methodologies and agile development methodologies respectively, are ranked higher than the second node, which is directly relevant to the input question, mentioning 'sobrecosto fuera del presupuesto estimado' as a main problem in software projects.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which mentions key problems in software projects, resulting in a flawless match.

Contextual Relevancy: 0.27 (Threshold: 0.5) - The score is 0.27 because the retrieval context mostly talks about unrelated topics such as estimation, indirect costs, agile software development, and company names, as stated in the reasons for irrelevancy, whereas only a few statements like 'costos directos', 'sobrecosto fuera del presupuesto estimado', and 'entregas tardias' are relevant to the input, but not enough to significantly impact the score.

## Caso de Prueba: 55 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre programación y desarrollo de software en términos de complejidad y escala de proyectos?

### Respuesta Generada:

Según el Documento 6, la principal diferencia entre programación y desarrollo de software se refiere a la complejidad y escala de proyectos. La programación se enfoca en la creación de código para resolver un problema o implementar una funcionalidad específica, mientras que el desarrollo de software implica la creación de sistemas complejos que involucran muchos participantes, múltiples tecnologías y requieren mantenimiento y actualizaciones constantes. En términos de complejidad, el desarrollo de software implica la integración de múltiples componentes, sistemas y tecnologías, lo que aumenta su complejidad. Además, los sistemas de software suelen ser más grandes y complejos que los programas individuales, lo que requiere una mayor planificación, diseño y estructura para garantizar su mantenibilidad y escalabilidad. En cuanto a la escala de proyectos, el desarrollo de software generalmente implica proyectos más grandes y complejos, con múltiples roles y responsabilidades, lo que requiere una mayor coordinación y comunicación entre los miembros del equipo de desarrollo. Por otro lado, la programación puede involucrar proyectos más pequeños y específicos, con un enfoque más estrecho en la implementación de una funcionalidad o solución particular. En resumen, la principal diferencia entre programación y desarrollo de software es que el desarrollo de software implica la creación de sistemas complejos y escalables que requieren una mayor planificación, diseño y estructura, mientras que la programación se enfoca en la creación de código para resolver un problema específico.

### Respuesta Esperada:

La principal diferencia entre programación y desarrollo de software en términos de complejidad y escala de proyectos es que la programación se enfoca en aplicaciones pequeñas, con una vida corta, y normalmente construidas desde cero, con uno o dos desarrolladores y pocos participantes, y requiere mantenimiento mínimo.

### Contexto Relevante:

1. Conceptos Programación Vs Desarrollo de software La programación es el proceso de trasladar un problema de su ambiente físico a un lenguaje que el computador pueda entender y obedecer. Uno o dos desar.....

### Contexto Recuperado:

1. diferencias escala procesos costo calendario tiempos habilidades y equipos de desarrollo tecnología riesgos aplicaciones web por que necesitamos una arquitectura necesitamos una arquitectura de softwa.....

2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 mentioning 'agil es un termino' and node 3 mentioning 'desarrollo de software agil', are correctly ranked lower than the relevant node 1, which explicitly discusses the differences in complexity and scale between programming and software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in the retrieval context, which shares similar concepts like 'desarrollo de software' and 'aplicaciones', resulting in a flawless match.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context mostly talks about agile development, its characteristics, and misconceptions, which are 'not directly related to the difference between programming and software development in terms of complexity and scale of projects', as stated in the reasons for irrelevancy.



## Caso de Prueba: 56 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre la programación y el desarrollo de software en términos de complejidad y alcance del proyecto?

### Respuesta Generada:

La principal diferencia entre la programación y el desarrollo de software se refiere a la complejidad y alcance del proyecto. La programación se enfoca en la escritura de código para resolver un problema específico o cumplir con un requisito dado, mientras que el desarrollo de software implica un enfoque más amplio y sistemático que considera todos los aspectos del ciclo de vida de un proyecto, desde el análisis de requisitos hasta la entrega y mantenimiento del software. En términos de complejidad, el desarrollo de software implica la integración de múltiples componentes, tecnologías y equipo de desarrollo, lo que requiere una mayor coordinación y comunicación entre los miembros del equipo. Además, el desarrollo de software considera aspectos como la reutilización de código, la escalabilidad, la mantenibilidad y la eficiencia, lo que la hace más compleja que la programación. En cuanto al alcance del proyecto, el desarrollo de software implica una visión más amplia que considera la arquitectura del sistema, la planificación y el seguimiento del proyecto, la gestión de riesgos y la entrega de valor al cliente. En cambio, la programación se enfoca en resolver un problema específico o implementar una funcionalidad determinada. En resumen, la principal diferencia entre la programación y el desarrollo de software es que el desarrollo de software implica una mayor complejidad y alcance del proyecto, requiriendo una visión más amplia y sistemática que considera todos los aspectos del ciclo de vida del proyecto.

### Respuesta Esperada:

La principal diferencia entre la programación y el desarrollo de software en términos de complejidad y alcance del proyecto es que la programación se enfoca en aplicaciones pequeñas, con una vida corta, y normalmente construidas desde cero, con mantenimiento mínimo, y solo requiere uno o dos desarrolladores, mientras que el desarrollo de software implica proyectos más complejos y de mayor alcance.

### Contexto Relevante:

1. Conceptos Programación Vs Desarrollo de software La programación es el proceso de trasladar un problema de su ambiente físico a un lenguaje que el computador pueda entender y obedecer. Uno o dos desar.....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

2. ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboración en equipo la planificación continua y el aprendizaje.....

3. nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor metodologías ágiles 12 principios definidos en el manifiesto ágil aceptamos que los requir.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and relevant answer!

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the 1st node which mentions software functionality, the 2nd node which explains agile development, and the 3rd node which discusses agile methodologies, are irrelevant nodes that do not address the input's question about the difference between programming and software development in terms of complexity and project scope.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output could be attributed to any node in the retrieval context, resulting in a complete lack of matching information.

Contextual Relevancy: 0.11 (Threshold: 0.5) - The score is 0.11 because the retrieval context does not mention the main difference between programming and software development, and only provides irrelevant information about agile development's characteristics, as stated in the reasons for irrelevancy.

## Caso de Prueba: 57 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre la programación y el desarrollo de software en términos de complejidad y escala de proyectos?

### Respuesta Generada:

Basándome en el contexto proporcionado, puedo responder que la principal diferencia entre la programación y el desarrollo de software en términos de complejidad y escala de proyectos radica en la naturaleza y envergadura de los proyectos que se abordan. La programación se enfoca en la creación de pequeñas aplicaciones o piezas de código que realizan tareas específicas, mientras que el desarrollo de software implica la construcción de sistemas complejos y escalables que requieren la integración de múltiples componentes, tecnologías y equipos de desarrollo. En términos de complejidad, el desarrollo de software implica considerar factores como la arquitectura de la aplicación, la escalabilidad, la mantenibilidad, la eficiencia y la interacción entre los componentes, lo que requiere una visión más amplia y una aproximación más sistemática. Por otro lado, la programación se centra en resolver problemas específicos y crear soluciones más puntualizadas. En cuanto a la escala de proyectos, el desarrollo de software puede abarcar proyectos de gran envergadura que requieren la coordinación de múltiples equipos, stakeholders y tecnologías, lo que implica una mayor complejidad y riesgo. En contraste, la programación suele implicar proyectos más pequeños y focalizados en una tarea específica. En resumen, la principal diferencia entre la programación y el desarrollo de software radica en la complejidad y la escala de los proyectos, con el desarrollo de software implicando proyectos más grandes, complejos y escalables que requieren una visión más amplia y una aproximación más sistemática. Este concepto se encuentra reflejado en el Documento 5, que destaca la diferencia entre la programación y la ingeniería de software, y en el Documento 3, que habla sobre la importancia de la arquitectura de software en la construcción de aplicaciones escalables y mantenibles.

### Respuesta Esperada:

La principal diferencia entre la programación y el desarrollo de software en términos de complejidad y escala de proyectos es que la programación se enfoca en aplicaciones pequeñas, con una vida corta, construidas desde cero, con mantenimiento mínimo y con la participación de uno o pocos desarrolladores, mientras que el desarrollo de software implica proyectos más complejos y a gran escala.

### Contexto Relevante:

1. Conceptos Programación Vs Desarrollo de software La programación es el proceso de trasladar un problema de su ambiente físico a un lenguaje que el computador pueda entender y obedecer. Uno o dos desar.....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. diferencias escala procesos costo calendario tiempos habilidades y equipos de desarrollo tecnologia riesgos aplicaciones web por que necesitamos una arquitectura necesitamos una arquitectura de softwa.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the output mostly addresses the input question, but gets sidetracked by mentioning a document, which isn't crucial to explaining the difference between programming and software development.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts, specifically the third node, is relevant to the input, as it mentions differences in scale, cost, calendar, times, skills, and teams of development, which is related to the complexity and scale of projects, and helps to understand the difference between programming and software development. The first and second nodes, ranked 1 and 2, are irrelevant, as they do not provide any information about the difference between programming and software development in terms of complexity and scale of projects.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 3rd node in retrieval context, which mentions differences in scale, directly supporting the complexity and scale of projects in the sentence.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because most of the retrieval context statements were irrelevant to the input, as they discussed topics like progress measurement, agile processes, technical excellence, simplicity, team organization, and team reflection, which are not related to the difference between programming and software development in terms of complexity and scale of projects, as stated in the input.

## Caso de Prueba: 58 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre programación y desarrollo de software en el contexto de crear aplicaciones?

### Respuesta Generada:

Según el Documento 1, la principal diferencia entre programación y desarrollo de software en el contexto de crear aplicaciones es la aplicación de conceptos, métodos y herramientas para crear aplicaciones de software. La programación se enfoca solo en escribir programas, mientras que el desarrollo de software implica un proceso estructurado que incluye el ciclo de vida del desarrollo de software, la ingeniería de software, la creación de aplicaciones de software de calidad, considerando aspectos como el diseño, la integración, los requisitos funcionales y no funcionales, entre otros. En otras palabras, la programación se centra en escribir código, mientras que el desarrollo de software abarca un proceso más amplio que implica la planificación, el diseño, la implementación, la prueba, la depuración, la documentación y el mantenimiento de software, con el fin de crear aplicaciones de alta calidad que satisfagan las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

La principal diferencia entre programación y desarrollo de software en el contexto de crear aplicaciones es que la programación se enfoca en traducir un problema a un lenguaje que el computador pueda entender, mientras que el desarrollo de software implica un proceso más amplio que abarca la creación de aplicaciones completas, lo que puede involucrar a más personas y etapas.

### Contexto Relevante:

1. Conceptos Programación Vs Desarrollo de software La programación es el proceso de trasladar un problema de su ambiente físico a un lenguaje que el computador pueda entender y obedecer. Uno o dos desar.....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

2. aplicar conocimiento la ingeniería de software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad a l t a d e m.....

3. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node directly addressing the differences between programming and software development, the second node explaining the application of software engineering principles, and the third node defining software engineering as a discipline, all of which are highly relevant to the input question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which discusses the software development process, resulting in a flawless match.

Contextual Relevancy: 0.21 (Threshold: 0.5) - The score is 0.21 because the retrieval context is mostly filled with statements about course presentation, definitions, software development life cycle, and importance of software, which are 'not directly related to the difference between programming and software development' as stated in the reasons for irrelevancy.

## **Caso de Prueba: 59 - Fallo**

### **Input del Usuario:**

¿Cómo abordan los equipos de desarrolladores con múltiples roles la complejidad de los sistemas complejos, garantizando su mantenimiento y escalabilidad en un entorno de crecimiento infinito y obsolescencia tecnológica constante?

### **Respuesta Generada:**

Para abordar la complejidad de los sistemas complejos, los equipos de desarrolladores con múltiples roles pueden utilizar varias estrategias para garantizar su mantenimiento y escalabilidad en un entorno de crecimiento infinito y obsolescencia tecnológica constante. En primer lugar, es fundamental implementar una arquitectura de software adecuada que permita la separación de responsabilidades, la escalabilidad y el mantenimiento. Esto se logra mediante la organización del sistema en capas, módulos o servicios independientes (Documento 5), lo que facilita la actualización y el mantenimiento de cada componente sin afectar la estabilidad del sistema. Además, los equipos de desarrolladores pueden utilizar prácticas ágiles como Scrum, que permite una mayor flexibilidad y adaptabilidad en la respuesta a los cambios y requisitos en constante evolución (Documento 2 y Documento 6). Scrum también facilita la comunicación entre los miembros del equipo y con los stakeholders a través de reuniones de sincronización y planificación regular. La reutilización de componentes y la modularidad también son clave para reducir costos y aumentar la eficiencia en el desarrollo y mantenimiento de sistemas complejos (Documento 1). De esta manera, los equipos pueden concentrarse en desarrollar nuevas funcionalidades y características en lugar de reinventar componentes ya existentes. La estimación de historias de usuario y la planificación ágil también permiten a los equipos de desarrolladores priorizar y enfocar sus esfuerzos en las características y funcionalidades más importantes y críticas para el negocio y los usuarios (Documento 4). Finalmente, la comunicación efectiva y la colaboración entre los miembros del equipo y con los stakeholders son fundamentales para asegurar que se entiendan los requisitos y se desenvuelvan soluciones que satisfacen las necesidades del cliente (Documento 6 y Documento 7). En resumen, los equipos de desarrolladores con múltiples roles pueden abordar la complejidad de los sistemas complejos mediante la implementación de arquitecturas de software escalables, la utilización de prácticas ágiles, la reutilización de componentes, la planificación y estimación ágil, y la comunicación efectiva y colaboración entre los miembros del equipo y con los stakeholders.

### **Respuesta Esperada:**

Los equipos de desarrolladores con múltiples roles abordan la complejidad de los sistemas complejos mediante la reutilización de componentes y la implementación de nuevas tecnologías para evitar la obsolescencia. Al mismo tiempo, garantizan el mantenimiento y escalabilidad de los sistemas a través de la colaboración entre los miembros del equipo con diferentes roles, lo que les permite abordar los desafíos de la complejidad desde diferentes perspectivas. De esta manera, pueden mantenerse al día con el crecimiento infinito y la obsolescencia tecnológica constante, reducir costos y garantizar la sostenibilidad de los sistemas complejos.

### **Contexto Relevante:**

1. Conceptos Programación Vs Ingeniería de software – Equipos de desarrolladores con múltiples roles. – Sistemas complejos. – Infinito crecimiento. – Muchos participantes. – Reutilización para reducir co.....

## Contexto Recuperado:

1. conceptos programacion vs ingenieria de software equipos de desarrolladores con multiples roles sistemas complejos infinito crecimiento muchos participantes reutilizacion para reducir costos requieren.....
2. pregunta en conferencia se reunen los representantes de los equipos por cada nivel nivel 1 5 veces nivel 2 2 veces y nivel 3 1 vez por semana duracion 15 30 y 45 minutos respectivamente preguntas durante.....
3. diferencias escala procesos costo calendario tiempos habilidades y equipos de desarrollo tecnologia riesgos aplicaciones web por que necesitamos una arquitectura necesitamos una arquitectura de softwa.....

## Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a comprehensive and relevant response without any irrelevant statements.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval context, which mentions key concepts like 'equipos de desarrolladores con multiples roles' and 'sistemas complejos', is correctly ranked higher than the second node, which is irrelevant and discusses meeting formats and team sizes. However, the score is not higher because the third node, which is relevant and discusses scalability and maintainability, is ranked lower than it should be.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions key concepts like 'equipos de desarrolladores con múltiples roles', 'sistemas complejos', 'reutilizacion para reducir costos', 'mantenimiento', and 'nuevas tecnologias para evitar obsolescencia', resulting in a flawless match.

Contextual Relevancy: 0.47 (Threshold: 0.5) - The score is 0.47 because the retrieval context is mostly irrelevant, with reasons such as 'The input asks about how development teams with multiple roles handle complex systems, but this statement is about the difference between programming concepts and software engineering.' and 'The statement is about meeting schedules and has no relation to handling complexity in systems.' However, some relevant statements were found, like 'equipos de desarrolladores con múltiples roles abordan sistemas complejos' and 'necesitamos una arquitectura de software al construir una app para organizar estructurar y optimizar su desarrollo asegurando escalabilidad mantenibilidad y eficiencia'.



## Caso de Prueba: 60 - Fallo

### Input del Usuario:

¿Cuál es el principal beneficio de reutilizar software en sistemas complejos con múltiples desarrolladores y participantes, en términos de reducción de costos y mantenimiento?

### Respuesta Generada:

Según el contexto proporcionado, el principal beneficio de reutilizar software en sistemas complejos con múltiples desarrolladores y participantes es reducir los costos. Al reutilizar software, se evita tener que desarrollar componentes o módulos desde cero, lo que ahorra tiempo, esfuerzo y recursos. Esto se traduce en una reducción de los costos directos y indirectos asociados con el desarrollo y mantenimiento del software. Además, la reutilización de software también contribuye a reducir los costos de mantenimiento, ya que los componentes reutilizados ya han sido probados y depurados, lo que reduce la probabilidad de errores y fallas. Esto significa que se necesitan menos recursos y esfuerzo para mantener el software, lo que reduce los costos de mantenimiento en el largo plazo. En resumen, la reutilización de software en sistemas complejos con múltiples desarrolladores y participantes es una estrategia efectiva para reducir los costos y el mantenimiento, lo que permite ahorrar tiempo, esfuerzo y recursos, y mejorar la eficiencia en el desarrollo y mantenimiento de software.

### Respuesta Esperada:

El principal beneficio de reutilizar software en sistemas complejos con múltiples desarrolladores y participantes es la reducción de costos y mantenimiento.

### Contexto Relevante:

1. Conceptos Programación Vs Ingeniería de software – Equipos de desarrolladores con múltiples roles. – Sistemas complejos. – Infinito crecimiento. – Muchos participantes. – Reutilización para reducir co.....

### Contexto Recuperado:

1. conceptos programacion vs ingenieria de software equipos de desarrolladores con multiples roles sistemas complejos infinito crecimiento muchos participantes reutilizacion para reducir costos requieren.....

2. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

3. interfaces de usuario interfaces de usuario es una secuencia de imagenes o prototipos de las interfaces de una app que muestran el conjunto de pantallas o interfaces de usuario de una aplicacion las c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node mentioning 'reutilizacion para reducir costos requieren mantenimiento', are correctly ranked higher than irrelevant nodes, like the second node discussing 'conceptos metodologias de desarrollo de software' and the third node about 'interfaces de usuario', which do not address the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, ensuring a flawless match.

Contextual Relevancy: 0.14 (Threshold: 0.5) - The score is 0.14 because most of the retrieval context statements, such as 'conceptos programacion vs ingenieria de software', 'equipos de desarrolladores con multiples roles', and others, do not address the benefits of reutilizing software in terms of cost reduction and maintenance, whereas only a few statements like 'reutilizacion para reducir costos' and 'requieren mantenimiento' are relevant to the input.

## Caso de Prueba: 61 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre la programación y la ingeniería de software en cuanto a la planificación y el mantenimiento a largo plazo de sistemas complejos?

### Respuesta Generada:

Según el contexto proporcionado, la principal diferencia entre la programación y la ingeniería de software en cuanto a la planificación y el mantenimiento a largo plazo de sistemas complejos es que la programación se enfoca en la creación de instrucciones codificadas para un lenguaje de programación, mientras que la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios, considerando la planificación, el diseño, el desarrollo, la implementación, las pruebas y el mantenimiento a largo plazo de sistemas complejos. En otras palabras, la programación se centra en la creación de código, mientras que la ingeniería de software se centra en la creación de sistemas complejos que requieren una perspectiva más amplia y una planificación a largo plazo. La ingeniería de software considera aspectos como la escalabilidad, la mantenibilidad, la eficiencia, la seguridad y la ética, entre otros, para crear sistemas complejos que se puedan mantener y actualizar con el tiempo, mientras que la programación se enfoca en la creación de un producto de software que funciona en un momento dado. En resumen, la principal diferencia es que la ingeniería de software considera la planificación y el mantenimiento a largo plazo de sistemas complejos como una parte integral del desarrollo de software, mientras que la programación se enfoca en la creación de código sin considerar necesariamente la planificación y el mantenimiento a largo plazo.

### Respuesta Esperada:

La principal diferencia entre la programación y la ingeniería de software en cuanto a la planificación y el mantenimiento a largo plazo de sistemas complejos es que la programación se enfoca en el desarrollo de código para resolver un problema específico, mientras que la ingeniería de software se centra en el diseño y planificación a largo plazo de sistemas complejos, considerando la reutilización, el crecimiento y el mantenimiento, con el fin de reducir costos y evitar la obsolescencia.

### Contexto Relevante:

1. Conceptos Programación Vs Ingeniería de software – Equipos de desarrolladores con múltiples roles. – Sistemas complejos. – Infinito crecimiento. – Muchos participantes. – Reutilización para reducir co.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a l ta d e m.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the output partially addresses the input question, but is brought down by its narrow focus on coding aspects and short-term functionality, failing to fully explore the differences in planning and long-term maintenance.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node, which clearly defines software engineering, and the third-ranked node, which highlights the importance of software engineering, are correctly ranked higher than the second-ranked node, which does not provide any relevant information, showing a strong contextual precision.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which discusses the systematic creation of high-quality software, aligning with the mentioned long-term planning and maintenance of complex systems.

Contextual Relevancy: 0.18 (Threshold: 0.5) - The score is 0.18 because the retrieval context mostly talks about the definition of software, which is not directly related to the planning and long-term maintenance of complex systems, as stated in the reasons for irrelevancy, such as 'The statement is about the definition of software, which is not directly related to the planning and long-term maintenance of complex systems.'

## Caso de Prueba: 62 - Fallo

### Input del Usuario:

¿Cuál es la mejor estrategia para evitar la obsolescencia en sistemas complejos desarrollados por equipos de desarrolladores con múltiples roles, donde la reutilización de código es clave para reducir costos?

### Respuesta Generada:

Según el Documento 1, la mejor estrategia para evitar la obsolescencia en sistemas complejos desarrollados por equipos de desarrolladores con múltiples roles es implementar nuevas tecnologías. Esta estrategia permite mantener el sistema actualizado y competitivo, evitando la obsolescencia y garantizando la escalabilidad y mantenibilidad del sistema. Además, la reutilización de código es clave para reducir costos, como se menciona en el Documento 1. Esto se logra mediante la implementación de una arquitectura de software adecuada, que permita la separación de responsabilidades y la reutilización de componentes, como se menciona en el Documento 4. Por otro lado, la estrategia de despliegue blue-green mencionada en el Documento 2 también puede ser útil para reducir el riesgo de obsolescencia. Esta estrategia implica duplicar temporalmente la infraestructura y desplegar todo a la vez, lo que permite minimizar el impacto de los cambios en el sistema y reducir el riesgo de errores. En resumen, la mejor estrategia para evitar la obsolescencia en sistemas complejos es implementar nuevas tecnologías, reutilizar código y implementar estrategias de despliegue como la estrategia blue-green.

### Respuesta Esperada:

La mejor estrategia para evitar la obsolescencia en sistemas complejos desarrollados por equipos de desarrolladores con múltiples roles, donde la reutilización de código es clave para reducir costos, es implementar una cultura de mantenimiento y actualización continua, incorporando nuevas tecnologías y herramientas que permitan la evolución del sistema sin afectar su funcionalidad. Esto se logra mediante la planificación y el diseño de sistemas modulares y escalables, que permitan la reutilización de código y la fácil integración de nuevas tecnologías, lo que reduce los costos y minimiza el riesgo de obsolescencia.

### Contexto Relevante:

1. Conceptos Programación Vs Ingeniería de software – Equipos de desarrolladores con múltiples roles. – Sistemas complejos. – Infinito crecimiento. – Muchos participantes. – Reutilización para reducir co.....

### Contexto Recuperado:

1. conceptos programacion vs ingenieria de software equipos de desarrolladores con multiples roles sistemas complejos infinito crecimiento muchos participantes reutilizacion para reducir costos requieren.....

2. eisc despliegue de aplicaciones web estrategias para actualizar el servidor despliegue bluegreen duplicar temporalmente la infraestructura y desplegar todo a la vez esta estrategia tambien se puede ap.....

3. interfaces de usuario la interfaz de usuario ui en una aplicacion es el punto de interaccion visual a traves del cual los usuarios se conectan y utilizan las funcionalidades del software incluye todos.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a highly relevant and comprehensive response.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 discussing deployment strategies and node 3 talking about user interfaces, are correctly ranked lower than the highly relevant node 1 that directly addresses the question.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the output partially matches the retrieval context, specifically sentence 1 is attributed to node 1 with 'reutilizacion para reducir costos' and 'requieren mantenimiento nuevas tecnologias para evitar obsolescencia', but the rest of the output lacks supporting evidence from the nodes in retrieval context.

Contextual Relevancy: 0.27 (Threshold: 0.5) - The score is 0.27 because the retrieval context mostly contains irrelevant statements that do not provide information about avoiding obsolescence in complex systems, such as 'conceptos programacion vs ingenieria de software' and 'infinito crecimiento muchos participantes', and only a few relevant statements like 'sistemas complejos desarrollados por equipos de desarrolladores con múltiples roles' and 'reutilización de código es clave para reducir costos'.

## Caso de Prueba: 63 - Fallo

### Input del Usuario:

¿Cuál es el papel del arquitecto de software en un equipo de desarrollo con múltiples roles, y cómo contribuye a garantizar la escalabilidad y mantenimiento de sistemas complejos?

### Respuesta Generada:

Según el Documento 6, el papel del arquitecto de software en un equipo de desarrollo con múltiples roles es clave para garantizar la escalabilidad y mantenimiento de sistemas complejos. La arquitectura de software se encarga de organizar, estructurar y optimizar el desarrollo del sistema, lo que asegura la escalabilidad, mantenibilidad y eficiencia del sistema. El arquitecto de software contribuye de la siguiente manera: \* Define la estructura del sistema, lo que facilita la separación de responsabilidades y mejora el rendimiento. \* Establece los principios rectores del diseño y evolución del sistema. \* Describe las múltiples facetas de la estructura del software, el comportamiento, el diseño, la codificación y la documentación. \* Proporciona una visión de trabajo para los desarrolladores, encapsulando las características importantes del diseño del sistema. En resumen, el arquitecto de software garantiza la escalabilidad y mantenimiento de sistemas complejos al definir una estructura clara y organizada para el desarrollo del sistema, lo que facilita la colaboración en equipo, la planificación continua y el aprendizaje continuo.

### Respuesta Esperada:

El papel del arquitecto de software en un equipo de desarrollo con múltiples roles es garantizar la escalabilidad y mantenimiento de sistemas complejos. Para lograr esto, el arquitecto de software debe diseñar y planificar la estructura general del sistema, teniendo en cuenta el infinito crecimiento y la necesidad de reutilización para reducir costos. Debe también considerar la obsolescencia y la necesidad de implementar nuevas tecnologías para mantener el sistema actualizado. Además, debe trabajar en estrecha colaboración con otros miembros del equipo, como desarrolladores y testers, para asegurar que el sistema sea escalable, mantenible y fácil de actualizar.

### Contexto Relevante:

1. Conceptos Programación Vs Ingeniería de software – Equipos de desarrolladores con múltiples roles. – Sistemas complejos. – Infinito crecimiento. – Muchos participantes. – Reutilización para reducir co.....

### Contexto Recuperado:

1. de software arquitectura a b c una arquitectura es la organizacion fundamental de un sistema encarnada en sus componentes sus relaciones entre si y con el medio en que se desarrolla asi como los princ.....

2. conceptos programacion vs ingenieria de software equipos de desarrolladores con multiples roles sistemas complejos infinito crecimiento muchos participantes reutilizacion para reducir costos requieren.....

3. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a precise and comprehensive answer to the question!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the 1st node mentioning 'arquitectura de software' and the 2nd node discussing key concepts like 'sistemas complejos', are ranked higher than the irrelevant 3rd node that only mentions 'agil'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be attributed to the 2nd node in the retrieval context, which perfectly aligns with the software architect's role in ensuring scalability and maintainability of complex systems.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because the retrieval context has many irrelevant statements that do not mention the role of a software architect or its contribution to scalability and maintenance of complex systems, as stated in the reasons for irrelevancy, such as 'a b c', 'i', 'infinite growth', 'many participants', 'reusing to reduce costs', and 'new technologies to avoid obsolescence', which are not related to the input, but has some relevant statements that mention 'arquitectura', 'sistemas complejos', and 'equipos de desarrolladores con multiples roles'.



## Caso de Prueba: 64 - Fallo

### Input del Usuario:

¿Cuáles son los beneficios clave de combinar prácticas ágiles y DevOps con inteligencia artificial para automatizar el ciclo de vida de desarrollo de software?

### Respuesta Generada:

Según el contexto proporcionado, especialmente en el Documento 4, combinar prácticas ágiles y DevOps con inteligencia artificial (IA) puede proporcionar varios beneficios clave para automatizar el ciclo de vida de desarrollo de software. Algunos de estos beneficios son: 1. **Automatización de partes del ciclo de vida**: La IA puede automatizar partes del ciclo de vida de desarrollo de software, como despliegue, pruebas y parte de la codificación, lo que puede mejorar la eficiencia y reducir los tiempos de entrega. 2. **Ingeniería de software aumentada**: La combinación de prácticas ágiles y DevOps con IA puede aumentar la ingeniería de software, permitiendo la creación de software más confiable, eficiente y escalable. 3. **mejora en la calidad del software**: La automatización y la inteligencia artificial pueden ayudar a mejorar la calidad del software, detectando errores y problemas de manera más eficiente y reduciendo la carga de trabajo manual. 4. **Optimización de recursos**: La automatización y la inteligencia artificial pueden optimizar los recursos disponibles, reducir costos y mejorar la velocidad de entrega del software. 5. **Mejora en la experiencia del usuario**: La combinación de prácticas ágiles y DevOps con IA puede mejorar la experiencia del usuario, permitiendo la creación de software más rápido, más confiable y más adaptable a las necesidades cambiantes de los usuarios. En resumen, la combinación de prácticas ágiles y DevOps con inteligencia artificial puede proporcionar beneficios significativos en términos de automatización, eficiencia, calidad y experiencia del usuario en el ciclo de vida de desarrollo de software.

### Respuesta Esperada:

Los beneficios clave de combinar prácticas ágiles y DevOps con inteligencia artificial para automatizar el ciclo de vida de desarrollo de software son la automatización de todas las etapas ágiles y la ingeniería de software aumentada con IA, lo que permite una mayor eficiencia y eficacia en el desarrollo de software.

### Contexto Relevante:

1. Conceptos Lo novedoso: ágil + DevOps + IA Automatización de partes del ciclo de vida (Despliegue, Pruebas y Parte de la codificación) Automatización de todas las etapas Ágil + Ingeniería de software .....

### Contexto Recuperado:

1. estimacion de tiempos y costos en marco de las metodologias agiles costo recursos invertidos en el desarrollo de la aplicacion precio cantidad de dinero que un usuario paga para acceder a la aplicacio.....
2. como iniciar el desarrollo de proyecto de software usando practicas agiles resumen...
3. uso de las metodologias agiles fuente annual state of agile report versionone medicion del exito de uso de las practicas agiles...

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all the top-ranked nodes in the retrieval contexts, including the 1st node about 'estimacion de tiempos y costos', the 2nd node about 'como iniciar el desarrollo de proyecto de software', and the 3rd node about 'uso de las metodologias agiles', are irrelevant to the input, failing to provide information about combining agile and DevOps with AI for automating the software development life cycle.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the expected output does not mention any specific details related to the nodes in retrieval context, resulting in no attributions to any node.

Contextual Relevancy: 0.71 (Threshold: 0.5) - The score is 0.71 because the retrieval context only partially addresses the input, focusing on agile methodologies and their measurements, but lacks direct connections to DevOps and AI for automating the software development lifecycle, as stated in the reasons for irrelevancy.

## Caso de Prueba: 65 - Éxito

### Input del Usuario:

¿Cómo puede la automatización de partes del ciclo de vida, como despliegue, pruebas y codificación, mediante el uso de tecnologías de IA, mejorar la eficiencia y la velocidad en el desarrollo ágil?

### Respuesta Generada:

Según el contexto proporcionado, la automatización de partes del ciclo de vida del desarrollo de software, como despliegue, pruebas y codificación, mediante el uso de tecnologías de IA, puede mejorar la eficiencia y la velocidad en el desarrollo ágil de la siguiente manera: \* Automatización del despliegue: mediante el uso de tecnologías de IA, el despliegue de aplicaciones puede ser automatizado, lo que reduce el tiempo y el esfuerzo humano necesarios para realizar este proceso. Esto permite una entrega más rápida y frecuente de software, lo que se alinea con los principios del desarrollo ágil. \* Automatización de pruebas: la automatización de pruebas mediante el uso de tecnologías de IA puede ayudar a identificar y corregir errores más rápido y de manera más eficiente. Esto reduce el tiempo y el esfuerzo necesarios para realizar pruebas, lo que permite una entrega más rápida y confiable del software. \* Automatización de la codificación: la automatización de la codificación mediante el uso de tecnologías de IA puede ayudar a reducir el tiempo y el esfuerzo necesarios para escribir código. Esto permite una entrega más rápida y eficiente del software, lo que se alinea con los principios del desarrollo ágil. Además, la automatización de partes del ciclo de vida del desarrollo de software puede mejorar la eficiencia y la velocidad en el desarrollo ágil de otras maneras, como: \* Reducir la cantidad de errores humanos: la automatización puede reducir la cantidad de errores humanos que se pueden producir durante el desarrollo de software, lo que mejora la calidad del software y reduce el tiempo y el esfuerzo necesarios para corregir errores. \* Incrementar la velocidad de entrega: la automatización puede permitir una entrega más rápida del software, lo que se alinea con los principios del desarrollo ágil. \* Mejorar la colaboración: la automatización puede ayudar a mejorar la colaboración entre los miembros del equipo de desarrollo, lo que permite una entrega más rápida y eficiente del software. En resumen, la automatización de partes del ciclo de vida del desarrollo de software, como despliegue, pruebas y codificación, mediante el uso de tecnologías de IA, puede mejorar la eficiencia y la velocidad en el desarrollo ágil al reducir el tiempo y el esfuerzo necesarios para realizar estas tareas, reducir la cantidad de errores humanos, incrementar la velocidad de entrega y mejorar la colaboración entre los miembros del equipo de desarrollo.

### Respuesta Esperada:

La automatización de partes del ciclo de vida, como despliegue, pruebas y codificación, mediante el uso de tecnologías de IA, puede mejorar la eficiencia y la velocidad en el desarrollo ágil al reducir el tiempo y el esfuerzo humano necesario en estas etapas. Al automatizar estas tareas, se pueden eliminar errores humanos y aumentar la precisión, lo que a su vez permite a los desarrolladores centrarse en tareas más complejas y de alto valor. Además, la automatización con IA puede permitir una mayor frecuencia de despliegue y pruebas, lo que se traduce en una mayor velocidad en el desarrollo y entrega de software.

### Contexto Relevante:

1. Conceptos Lo novedoso: ágil + DevOps + IA Automatización de partes del ciclo de vida (Despliegue, Pruebas y Parte de la codificación) Automatización de todas las etapas Ágil + Ingeniería de software  
.....

### **Contexto Recuperado:**

1. conceptos lo novedoso agil devops ia automatizacion de partes del ciclo de vida despliegue pruebas y parte de la codificacion automatizacion de todas las etapas agil ingenieria de software aumentad.....
2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy dia interactuar con los stake.....
3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, directly address the input by mentioning relevant concepts such as automation of parts of the cycle of life using IA, software development life cycle, and specific stages of the software development life cycle.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions automation of parts of the lifecycle and its benefits, resulting in a seamless match.

Contextual Relevancy: 0.65 (Threshold: 0.5) - The score is 0.65 because although the retrieval context mentions relevant terms like 'ia automatizacion de partes del ciclo de vida', 'despliegue pruebas y parte de la codificacion', and 'devops', the majority of the reasons for irrelevancy highlight the lack of direct relation to automating parts of the development cycle using AI technologies, as stated in the input.

## Caso de Prueba: 66 - Fallo

### Input del Usuario:

¿Cuál es el papel de la automatización en la integración de pruebas en un entorno de desarrollo ágil y DevOps?

### Respuesta Generada:

Según el documento 2, la automatización juega un papel fundamental en la integración de pruebas en un entorno de desarrollo ágil y DevOps. La automatización de partes del ciclo de vida, como el despliegue, pruebas y parte de la codificación, es un concepto novedoso en la integración de Agile y DevOps. En particular, la automatización de pruebas es una práctica clave en la integración de pruebas en un entorno de desarrollo ágil y DevOps. La automatización de pruebas permite ejecutar pruebas de forma rápida y repetida, lo que ayuda a asegurar la calidad del software y a detectar errores temprano en el ciclo de desarrollo. En el contexto de Scrum, la automatización de pruebas es una parte integral del Definition of Done (DoD), como se menciona en el documento 1. La DoD establece que se han creado pruebas unitarias y todas las pruebas son satisfactorias, lo que garantiza que el software cumpla con los requisitos esperados. Además, la automatización de pruebas también se menciona en los documentos 5 y 10, donde se habla sobre la importancia de la codificación, pruebas de software y despliegue en el ciclo de vida del desarrollo de software. En resumen, la automatización juega un papel crucial en la integración de pruebas en un entorno de desarrollo ágil y DevOps, permitiendo la ejecución rápida y repetida de pruebas para asegurar la calidad del software y detectar errores temprano en el ciclo de desarrollo.

### Respuesta Esperada:

En un entorno de desarrollo ágil y DevOps, el papel de la automatización en la integración de pruebas es fundamental, ya que permite automatizar partes del ciclo de vida, como el despliegue, pruebas y parte de la codificación. De esta manera, se logra una mayor eficiencia y velocidad en el proceso de desarrollo, lo que se alinea con los principios ágiles y DevOps. La automatización de todas las etapas ágiles y la ingeniería de software aumentada con IA permiten una mayor precisión y reducción de errores en la integración de pruebas.

### Contexto Relevante:

1. Conceptos Lo novedoso: ágil + DevOps + IA Automatización de partes del ciclo de vida (Despliegue, Pruebas y Parte de la codificación) Automatización de todas las etapas Ágil + Ingeniería de software .....

### Contexto Recuperado:

1. practicas agiles en scrum dod definition of done criterio cumplido criterios de aceptacion cumplidos la historia de usuario cumple con todos los criterios de aceptacion definidos previamente codigo de.....

2. conceptos lo novedoso agil devops ia automatizacion de partes del ciclo de vida despliegue pruebas y parte de la codificacion automatizacion de todas las etapas agil ingenieria de software aumentad.....

3. practicas agiles en scrum  
definition of done  
criterio cumplido  
documentacion actualizada se ha actualizado la documentacion tecnica  
yo del usuario si es necesario  
codigo subido al repositorio el C.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.92 (Threshold: 0.5) - The score is 0.92 because the response is highly relevant to the input, but slightly deducted for the minor detour about documents, which, although related, doesn't directly address the role of automation in integration testing.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which explicitly mention the integration of tests and the role of automation, are ranked higher than the irrelevant node 3, which only mentions deployment in a test environment and is focused on Scrum and agile practices.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which mentions key concepts like agile, DevOps, automation, and IA, resulting in a seamless alignment.

Contextual Relevancy: 0.29 (Threshold: 0.5) - The score is 0.29 because the retrieval context mostly talks about unrelated topics such as 'practicas agiles en scrum', 'codigo desarrollado', 'revision de codigo', and others, with only a few statements mentioning 'pruebas' and 'automatizacion' which are somewhat related to the input.

## Caso de Prueba: 67 - Fallo

### Input del Usuario:

¿Cuáles son los beneficios clave de integrar la automatización con inteligencia artificial en el ciclo de vida de desarrollo de software, especialmente en las etapas de despliegue, Pruebas y codificación?

### Respuesta Generada:

Según el Documento 4, la integración de la automatización con inteligencia artificial (IA) en el ciclo de vida de desarrollo de software, especialmente en las etapas de despliegue, pruebas y codificación, puede ofrecer varios beneficios clave. Algunos de estos beneficios incluyen: 1. **\*\*Aumento de la eficiencia\*\***: La automatización de tareas repetitivas y tediosas puede liberar tiempo y recursos para que los desarrolladores se centren en tareas más complejas y de mayor valor. 2. **\*\*Mejora de la calidad\*\***: La IA puede ayudar a identificar y corregir errores y defectos en el código de manera más eficiente y precisa que los métodos manuales. 3. **\*\*Reducción de errores\*\***: La automatización de pruebas y despliegue puede reducir significativamente la probabilidad de errores humanos y defectos en el software. 4. **\*\*Incremento de la velocidad\*\***: La automatización y la IA pueden acelerar el proceso de desarrollo y despliegue de software, permitiendo una entrega más rápida y frecuente de nuevas características y actualizaciones. 5. **\*\*Mejora de la experiencia del usuario\*\***: La automatización y la IA pueden ayudar a personalizar y mejorar la experiencia del usuario, por ejemplo, mediante la recomendación de características o la identificación de patrones de comportamiento. 6. **\*\*Reducción de costos\*\***: La automatización y la IA pueden reducir los costos asociados con el desarrollo y mantenimiento de software, ya que requieren menos recursos humanos y de infraestructura. 7. **\*\*Escalabilidad\*\***: La automatización y la IA pueden ayudar a escalar el desarrollo y despliegue de software de manera más eficiente, lo que es especialmente importante para aplicaciones que requieren un gran número de usuarios concurrentes. En resumen, la integración de la automatización con inteligencia artificial en el ciclo de vida de desarrollo de software puede ofrecer una serie de beneficios clave, incluyendo la mejora de la eficiencia, la calidad y la velocidad, así como la reducción de errores y costos.

### Respuesta Esperada:

Los beneficios clave de integrar la automatización con inteligencia artificial en el ciclo de vida de desarrollo de software, especialmente en las etapas de despliegue, pruebas y codificación, son la automatización de partes del ciclo de vida, lo que permite un enfoque ágil y DevOps, y la ingeniería de software aumentada con IA.

### Contexto Relevante:

1. Conceptos Lo novedoso: ágil + DevOps + IA Automatización de partes del ciclo de vida (Despliegue, Pruebas y Parte de la codificación) Automatización de todas las etapas Ágil + Ingeniería de software .....

### Contexto Recuperado:

1. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

2. estimacion de tiempos y costos en marco de las metodologias agiles costo recursos invertidos en el desarrollo de la aplicacion precio cantidad de dinero que un usuario paga para acceder a la aplicacio.....

3. conceptos aspectos clave de la ingenieria de software 5 gestion de proyectos aplica tecnicas de gestion de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and comprehensive answer without any irrelevant information!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant, and it's ranked third. The first node, ranked 1, is about 'practicas agiles en scrum retrospectiva' which is not directly related to integrating automation with AI in the software development life cycle. The second node, ranked 2, is about 'estimacion de tiempos y costos' which is about agile methodologies and cost estimation, not about integrating automation with AI. The relevant node, ranked 3, mentions the software development life cycle, including phases like deployment, testing, and coding, which is relevant to integrating automation with AI.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context, indicating a complete mismatch.

Contextual Relevancy: 0.09 (Threshold: 0.5) - The score is 0.09 because the retrieval context is mostly about Scrum practices, project management, and general software engineering principles, which have no connection to integrating automation with AI in the software development lifecycle, as stated in the reasons for irrelevancy, such as 'The statement is about Scrum retrospective practices, which has no connection to integrating automation with AI in the software development lifecycle.'



## Caso de Prueba: 68 - Éxito

### Input del Usuario:

¿Cuál es el impacto de la inteligencia artificial en la ingeniería de software ágil y cómo puede mejorar la eficiencia y la calidad en el desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, la inteligencia artificial (IA) está revolucionando la ingeniería de software ágil. En el documento 4, se menciona que la IA puede automatizar partes del ciclo de vida de desarrollo de software, incluyendo el despliegue, pruebas y parte de la codificación. Esto puede llevar a una mayor eficiencia y calidad en el desarrollo de software. La IA puede mejorar la eficiencia en el desarrollo de software de varias maneras: 1. Automatización de tareas repetitivas: La IA puede automatizar tareas como pruebas, depuración y refactoring, lo que puede ahorrar tiempo y recursos. 2. Análisis de datos: La IA puede analizar grandes cantidades de datos para identificar patrones y tendencias, lo que puede ayudar a los desarrolladores a tomar decisiones informadas sobre el diseño y la implementación del software. 3. Identificación de problemas: La IA puede identificar problemas potenciales en el código y alertar a los desarrolladores, lo que puede ayudar a reducir los errores y mejorar la calidad del software. Además, la IA puede mejorar la calidad del software de varias maneras: 1. Mejora de la precisión: La IA puede ayudar a los desarrolladores a escribir código más preciso y eficiente. 2. Identificación de vulnerabilidades: La IA puede identificar vulnerabilidades de seguridad en el código y ayudar a los desarrolladores a abordarlas. 3. Mejora de la experiencia del usuario: La IA puede ayudar a los desarrolladores a crear software que sea más intuitivo y fácil de usar. En resumen, la inteligencia artificial puede tener un impacto significativo en la ingeniería de software ágil, mejorando la eficiencia y la calidad del desarrollo de software. Sin embargo, es importante mencionar que la IA no reemplaza la necesidad de habilidades y conocimientos en ingeniería de software, sino que se convierte en una herramienta valiosa para apoyar y mejorar el proceso de desarrollo.

### Respuesta Esperada:

El impacto de la inteligencia artificial en la ingeniería de software ágil es significativo, ya que puede mejorar la eficiencia y la calidad en el desarrollo de software. La automatización de partes del ciclo de vida, como el despliegue, pruebas y parte de la codificación, permite ahorrar tiempo y recursos, reducir errores y mejorar la precisión. Además, la automatización de todas las etapas ágiles permite una mayor velocidad y flexibilidad en el desarrollo de software. La ingeniería de software aumentada con IA también permite una mayor eficiencia en la toma de decisiones y la resolución de problemas, lo que conduce a una mayor calidad en el software final.

### Contexto Relevante:

1. Conceptos Lo novedoso: ágil + DevOps + IA Automatización de partes del ciclo de vida (Despliegue, Pruebas y Parte de la codificación) Automatización de todas las etapas Ágil + Ingeniería de software  
.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue.....
2. ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboración en equipo la planificación continua y el aprendizaje.....
3. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a comprehensive and accurate response without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which defines software engineering and its focus on creating high-quality software, and node 2 which explains the concept of agile development, are ranked higher than the irrelevant node 3, which only defines software but lacks information about AI's impact on agile software engineering.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly supported by the nodes in retrieval context, with mentions of 'ingeniería de software', 'automatización', 'despliegue', 'etapas ágiles', and 'toma de decisiones' all aligning with nodes 1, 2, and 3.

Contextual Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the retrieval context is partially relevant, as it mentions software engineering and agile development, but lacks direct connection to artificial intelligence, a key aspect of the input.

## Caso de Prueba: 69 - Fallo

### Input del Usuario:

¿Cuál es la mejor forma de interactuar con los stakeholders y usuarios para recopilar requerimientos efectivos en la etapa de ideación del ciclo de vida del desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, la mejor forma de interactuar con los stakeholders y usuarios para recopilar requerimientos efectivos en la etapa de ideación del ciclo de vida del desarrollo de software es a través de la elicitación de requisitos. La elicitación de requisitos es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. Esto implica la identificación de actores y funcionalidades, análisis de requerimientos, especificación y validación. Algunas técnicas efectivas para la elicitación de requisitos mencionadas en los documentos son: \* Prototipado de interfaz de usuario: crear prototipos de interfaz de usuario para validar los requisitos con los usuarios. \* Recorrido de BPM: crear un modelo de proceso de negocio que muestre todo el proceso para entender las necesidades de los usuarios. \* Historias de usuario: utilizar historias de usuario como una técnica para obtener requisitos desde la perspectiva de los usuarios mediante el modelado de procesos de negocio. \* Validación de expertos: tener personas con experiencia que revisen los requisitos y aprueben o rechacen el requerimiento. En resumen, la mejor forma de interactuar con los stakeholders y usuarios para recopilar requerimientos efectivos en la etapa de ideación del ciclo de vida del desarrollo de software es a través de la elicitación de requisitos, utilizando técnicas como prototipado de interfaz de usuario, recorrido de BPM, historias de usuario y validación de expertos.

### Respuesta Esperada:

La mejor forma de interactuar con los stakeholders y usuarios para recopilar requerimientos efectivos en la etapa de ideación del ciclo de vida del desarrollo de software es interactuar con ellos para entender los procesos y recopilar los requerimientos del proyecto.

### Contexto Relevante:

1. Ciclo de vida del desarrollo de software Lluvia de ideas que resuelven un problema particular que enfrentan los usuarios. IDEACIÓN Concepto: Ciclo de vida del software hoy día Interactuar con los stakeholders.....

### Contexto Recuperado:

1. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy dia interactuar con los stakeholders.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitacion es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
3. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por

los desarrollado.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are highly relevant to the input, providing direct quotes and insightful information about gathering effective requirements, making them perfectly ranked.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, ensuring a seamless match.

Contextual Relevancy: 0.41 (Threshold: 0.5) - The score is 0.41 because the retrieval context mostly talks about the software development life cycle, testing, and deploying, which are not directly related to interacting with stakeholders and users for effective requirement gathering, as stated in the input.

## Caso de Prueba: 70 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal del ciclo de vida del desarrollo de software en la etapa de requerimientos?

### Respuesta Generada:

Según el contexto proporcionado, el objetivo principal del ciclo de vida del desarrollo de software en la etapa de requerimientos es entender el problema o necesidad de los usuarios y escribir los requerimientos del sistema. Esto se menciona en el Documento 2 y se refiere a la etapa de análisis, donde se busca comprender las necesidades de los usuarios y plasmarlas en requerimientos funcionales y no funcionales. En otras palabras, el objetivo principal de esta etapa es obtener una comprensión clara de lo que se necesita desarrollar, qué problemas se deben resolver y qué características debe tener el sistema de software para satisfacer las necesidades de los usuarios. Esto se logra a través de la interacción con los stakeholders y usuarios, y la utilización de técnicas de elicitation para obtener los requerimientos del proyecto.

### Respuesta Esperada:

El objetivo principal del ciclo de vida del desarrollo de software en la etapa de requerimientos es interactuar con los stakeholders y usuarios para entender los procesos y recolectar los requerimientos del proyecto.

### Contexto Relevante:

1. Ciclo de vida del desarrollo de software Lluvia de ideas que resuelven un problema particular que enfrentan los usuarios. IDEACIÓN Concepto: Ciclo de vida del software hoy día Interactuar con los sta.....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

2. conceptos análisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseño es una descripción o especificación de proceso.....

3. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stake.....

### Métricas de Evaluación:

Answer Relevancy: 0.78 (Threshold: 0.5) - The score is 0.78 because the response is mostly relevant, but slightly derailed by mentioning a specific document and a different stage of the software development lifecycle.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the most relevant node (node 3) directly answering the question, and the less relevant nodes (nodes 1 and 2) ranked lower, as they only relate to the question without directly answering it.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly aligns with the 3rd node in retrieval context, which explicitly mentions the same information.

Contextual Relevancy: 0.29 (Threshold: 0.5) - The score is 0.29 because the retrieval context is mostly irrelevant, with statements like 'presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software' and 'aplicar conceptos metodos y herramientas para crear aplicaciones de software' that have no connection to the main objective of the software development life cycle in the requirements stage, and only a few relevant statements like 'ciclo de vida proceso de creacion de software proceso o marco estructurado para el desarrollo de software' and 'requerimientos' that are related to the input.

## Caso de Prueba: 71 - Fallo

### Input del Usuario:

¿Cuál es el papel del diseño en el ciclo de vida del software y cómo influye en la creación de un sistema de software?

### Respuesta Generada:

Según los documentos proporcionados, el diseño es una etapa fundamental en el ciclo de vida del desarrollo de software. En el Documento 2, se menciona que el diseño implica "crear la arquitectura de un sistema de software, es decir, sus componentes, sus interfaces y su forma de interactuar". En el Documento 3, se define el diseño como "una descripción o especificación de procesos, componentes e interfaces que representan una solución para el sistema a construir". En otras palabras, el diseño es la fase donde se define la estructura y la organización del sistema de software, incluyendo sus componentes, interfaces y relaciones entre sí. El papel del diseño en el ciclo de vida del software es crucial, ya que influye directamente en la creación del sistema de software. El diseño determina cómo se implementará el sistema, cuáles serán sus componentes y cómo se interactuarán entre sí. Un buen diseño permite crear un sistema de software que sea escalable, mantenible y que satisfaga las necesidades de los usuarios. Además, el diseño también influye en la calidad del sistema de software, ya que un diseño bien estructurado y fácil de entender puede reducir la complejidad del sistema y minimizar los errores. En el Documento 7, se destaca la importancia de la arquitectura del software, que es una visión de trabajo para los desarrolladores que encapsula las características importantes del diseño de un sistema. En resumen, el diseño es una etapa crítica en el ciclo de vida del desarrollo de software, ya que define la estructura y la organización del sistema de software y determina cómo se implementará el sistema. Un buen diseño es fundamental para crear un sistema de software de alta calidad que satisfaga las necesidades de los usuarios.

### Respuesta Esperada:

El papel del diseño en el ciclo de vida del software es crear la arquitectura de un sistema de software, es decir, definir sus componentes, interfaces y forma de interactuar. El diseño influye en la creación de un sistema de software ya que establece la base para la construcción del sistema, determinando cómo se van a implementar los requisitos recopilados en la etapa de ideación y requisitos. Un buen diseño permite la creación de un sistema de software escalable, mantenible y fácil de entender, lo que a su vez facilita el desarrollo, pruebas y despliegue del sistema.

### Contexto Relevante:

1. Ciclo de vida del desarrollo de software Lluvia de ideas que resuelven un problema particular que enfrentan los usuarios. IDEACIÓN Concepto: Ciclo de vida del software hoy día Interactuar con los sta.....

### Contexto Recuperado:

1. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy día interactuar con los stake.....

3. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a clear and concise answer to the question!

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first node, which does not explicitly address the role of design in the software development life cycle, is ranked higher than the second node, which clearly explains the role of design in creating the architecture of a system, defining its components, interfaces, and interactions. Additionally, the third node, which only explains the concept of design and its relation to the software development life cycle, is ranked higher than the second node.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 2nd node in retrieval context, showcasing a seamless connection between the output and the context.

Contextual Relevancy: 0.44 (Threshold: 0.5) - The score is 0.44 because the retrieval context contains many irrelevant statements, such as 'resumen', 'lluvia de ideas', and statements about testing, deployment, and monitoring, which are not related to the role of design in the software development life cycle, overshadowing the few relevant statements that actually discuss design and the software development lifecycle.



## Caso de Prueba: 72 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de la etapa de ideación en el ciclo de vida del software?

### Respuesta Generada:

Según el Documento 2, el propósito principal de la etapa de ideación en el ciclo de vida del software es interactuar con los stakeholders y usuarios para entender los procesos y recopilar los requisitos del proyecto. En esta etapa, se generan ideas que resuelven un problema particular que enfrentan los usuarios.

### Respuesta Esperada:

El propósito principal de la etapa de ideación en el ciclo de vida del software es interactuar con los stakeholders y usuarios para entender los procesos y recolectar los requerimientos del proyecto, es decir, generar ideas que resuelvan un problema particular que enfrentan los usuarios.

### Contexto Relevante:

1. Ciclo de vida del desarrollo de software Lluvia de ideas que resuelven un problema particular que enfrentan los usuarios. IDEACIÓN Concepto: Ciclo de vida del software hoy día Interactuar con los sta.....

### Contexto Recuperado:

1. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....
2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stake.....
3. presentación de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the response is mostly on topic, but includes some tangential information about interacting with stakeholders and understanding processes, which detracts from its overall relevance to the primary purpose of the ideation stage.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first node, which is irrelevant, is ranked higher than the second node, which is relevant and clearly states the main purpose of the ideation stage in the software development life cycle, and the third node, which is also irrelevant, is ranked lower than the second node.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, showcasing a seamless understanding of the software development lifecycle.

Contextual Relevancy: 0.29 (Threshold: 0.5) - The score is 0.29 because the retrieval context mostly talks about coding concepts, software testing, and development, which are not related to the purpose of the ideation stage in the software development life cycle, as stated in the reasons for irrelevancy.

## Caso de Prueba: 73 - Fallo

### Input del Usuario:

¿Cuál es el propósito de la fase de pruebas en el ciclo de vida del desarrollo de software?

### Respuesta Generada:

Según el documento 1, la fase de pruebas en el ciclo de vida del desarrollo de software es el proceso de evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer. En otras palabras, el propósito de la fase de pruebas es asegurarse de que el software cumpla con los requisitos y funcionalidades esperadas, encontrando y corrigiendo posibles errores o defectos en el proceso.

### Respuesta Esperada:

El propósito de la fase de pruebas en el ciclo de vida del desarrollo de software es evaluar la calidad del sistema para encontrar y corregir posibles errores.

### Contexto Relevante:

1. Ciclo de vida del desarrollo de software Lluvia de ideas que resuelven un problema particular que enfrentan los usuarios. IDEACIÓN Concepto: Ciclo de vida del software hoy día Interactuar con los sta.....

### Contexto Recuperado:

1. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

2. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de prácticas principios y procedimientos que guían y regulan el proceso de creación.....

3. conceptos aspectos clave de la ingeniería de software 5 gestión de proyectos aplica técnicas de gestión de proyectos para planificar organizar y controlar el desarrollo de software asegurando que se e.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question about the purpose of the testing phase in the software development life cycle, providing a clear and relevant response.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts directly addresses the purpose of the testing phase, and the third-ranked node also mentions 'pruebas' as one of the phases, but the second-ranked node, which talks about methodologies of software development, should be ranked lower as it is not directly related to the

purpose of the testing phase.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be confidently attributed to the 1st node in the retrieval context, which explicitly discusses the software development life cycle.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context mostly contains irrelevant information, such as 'conceptos la codificacion...' and statements that only describe the scope of the software development lifecycle or the goals of using a software development methodology, without providing any meaningful information about the purpose of the testing phase, as stated in the reasons for irrelevancy.

## Caso de Prueba: 74 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de una metodología de desarrollo de software?

### Respuesta Generada:

Según el Documento 2, el propósito principal de una metodología de desarrollo de software es "mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo". Esto se logra mediante un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de creación, implementación, despliegue y mantenimiento de sistemas de software.

### Respuesta Esperada:

El propósito principal de una metodología de desarrollo de software es mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Una metodología de desarrollo de software es un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de cre.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....
2. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....
3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant, but mentions some non-essential details about optimizing resources and minimizing risks, which aren't the primary purpose of a software development methodology.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the second node, which explicitly states the main purpose of a software development methodology, is correctly ranked higher than the first and third nodes, which do not directly address the main purpose and instead discuss agile development principles.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, indicating a flawless match.

Contextual Relevancy: 0.93 (Threshold: 0.5) - The score is 0.93 because the retrieval context provided relevant information about software development methodologies, such as 'una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos...' and 'estas metodologias establecen un marco para organizar y gestionar las tareas...', which aligns with the input question, despite containing some irrelevant information about a specific entity.

## Caso de Prueba: 75 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de una metodología de desarrollo de software en relación con el ciclo de vida del software?

### Respuesta Generada:

Según el Documento 1, el propósito principal de una metodología de desarrollo de software es establecer un marco para organizar y gestionar las tareas involucradas en cada fase del ciclo de vida del software, con el fin de mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo. En otras palabras, la metodología de desarrollo de software tiene como objetivo principal guiar y regular el proceso de creación, implementación, despliegue y mantenimiento de sistemas de software, a través de un conjunto estructurado de prácticas, principios y procedimientos, que abarcan todas las fases del ciclo de vida del software, desde la recolección de requisitos hasta el despliegue y soporte.

### Respuesta Esperada:

El propósito principal de una metodología de desarrollo de software en relación con el ciclo de vida del software es establecer un marco para organizar y gestionar las tareas involucradas en cada fase del ciclo de vida del software, con el fin de mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Una metodología de desarrollo de software es un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de cre.....

### Contexto Recuperado:

1. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

### Métricas de Evaluación:

Answer Relevancy: 0.70 (Threshold: 0.5) - The score is 0.70 because the output addresses the main purpose of a software development methodology in relation to the software life cycle, but gets

sidetracked with details about product quality, resource optimization, and risk minimization.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly addressing the question and irrelevant nodes, such as the second node about 'desarrollo de software agil' and the third node about 'presentacion de curso', correctly ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which aligns with the software development methodology's purpose of organizing and managing tasks in each phase of the software lifecycle.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context contained mostly irrelevant information, such as 'scrumcolombia desarrollo de software i' and 'presentacion de curso características del curso definiciones', which are not related to the main purpose of a software development methodology in relation to the software life cycle, and only a few relevant statements that briefly touch on the topic.



## Caso de Prueba: 76 - Éxito

### Input del Usuario:

¿Cuál es el primer paso en el ciclo de vida del software que se menciona en el contexto de las metodologías de desarrollo de software?

### Respuesta Generada:

Según el contexto proporcionado, el primer paso en el ciclo de vida del software es la "Ideación" o "Concepción", que se menciona en el Documento 1. En este paso, se busca entender los procesos y recopilar los requisitos del proyecto a través de la interacción con los stakeholders y usuarios, con el fin de resolver un problema particular que enfrentan los usuarios.

### Respuesta Esperada:

Según el contexto, el primer paso en el ciclo de vida del software es la recolección de requisitos.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Una metodología de desarrollo de software es un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de cre.....

### Contexto Recuperado:

1. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy día interactuar con los stake.....
2. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....
3. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant, but mentions a document, which is not directly related to the first step in the software development lifecycle, preventing a perfect score.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which clearly addresses the question, is correctly ranked higher than the second node, which is not directly related to the ciclo de vida del software and its steps, and the third node, which is also relevant to the topic, is ranked accordingly.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in the retrieval context, which explicitly mentions 'recolectar los requerimientos del proyecto', matching the expected output's first step in the software life cycle.

Contextual Relevancy: 0.58 (Threshold: 0.5) - The score is 0.58 because the retrieval context contains mostly irrelevant information such as 'lluvia de ideas', 'presentacion de curso', and 'aplicar conceptos metodos y herramientas para crear aplicaciones de software', which have no connection to the first step in the software development life cycle, but it does contain some relevant statements like 'ciclo de vida del desarrollo de software' and 'ideacion'.

## Caso de Prueba: 77 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal de utilizar una metodología de desarrollo de software en la gestión de proyectos?

### Respuesta Generada:

Según el contexto proporcionado, el objetivo principal de utilizar una metodología de desarrollo de software en la gestión de proyectos es mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo (Documento 3). Esto se logra estableciendo un marco para organizar y gestionar las tareas involucradas en cada fase del ciclo de vida del software, desde la recolección de requisitos hasta el despliegue y soporte. De esta manera, se busca entregar un software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

El objetivo principal de utilizar una metodología de desarrollo de software en la gestión de proyectos es mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Una metodología de desarrollo de software es un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de cre.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

2. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

3. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a spot-on answer without any irrelevant information!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts directly addresses the main objective of using a software development methodology in project management, with the third node being the relevant one, stating that the main objective is 'mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo'. The first and second nodes, ranked higher, are irrelevant as they mention 'desarrollo agil de software' and 'objetivos del curso' respectively, which do not directly relate to the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which matches the sentence exactly.

Contextual Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because the retrieval context is only partially relevant, as it contains some relevant statements about software development methodologies, but also includes irrelevant information about a specific organization and course objectives, as stated in the reasons for irrelevancy.

## Caso de Prueba: 78 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de establecer una metodología de desarrollo de software en cuanto a la calidad del producto final?

### Respuesta Generada:

Según el Documento 1, el propósito principal de establecer una metodología de desarrollo de software es "mejorar la calidad del producto final". Esto se logra mediante el establecimiento de un marco que guíe y regule el proceso de creación, implementación, despliegue y mantenimiento de sistemas de software, optimizando recursos y minimizando riesgos. En otras palabras, la metodología de desarrollo de software es fundamental para asegurar que el producto final sea de alta calidad y satisfaga las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

El propósito principal de establecer una metodología de desarrollo de software en cuanto a la calidad del producto final es mejorar la calidad del producto final.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Una metodología de desarrollo de software es un conjunto estructurado de prácticas, principios y procedimientos que guían y regulan el proceso de cre.....

### Contexto Recuperado:

1. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output addresses the primary purpose of establishing a software development methodology, but gets sidetracked by mentioning secondary benefits and unrelated goals.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which clearly states that the purpose of establishing a software development methodology is

to 'mejorar la calidad del producto final', are ranked higher than the irrelevant nodes, such as the second node that focuses on the agile paradigm and the third node that is about understanding project objectives, which should be ranked lower due to their lack of direct relation to the quality of the final product.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, specifically the 1st node, which ensures the quality of the final product.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context is mostly irrelevant, with statements talking about agile development being a paradigm, alternative, umbrella, or different way of managing teams, which is not directly related to the input, whereas only a few statements like 'una metodologia de desarrollo de software es un conjunto estructurado...' and 'con el fin de mejorar la calidad del producto final...' are relevant to the input.

## Caso de Prueba: 79 - Fallo

### Input del Usuario:

¿Cuál es la principal característica que define a la metodología de desarrollo de software conocida como Cascada?

### Respuesta Generada:

La principal característica que define a la metodología de desarrollo de software conocida como Cascada no se menciona explícitamente en los documentos proporcionados. Sin embargo, podemos inferir que no se está refiriendo a la metodología Cascada porque en los documentos se habla de metodologías ágiles, Scrum, y desarrollo de software ágil, enfatizando en la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo. La metodología Cascada se caracteriza por ser una metodología de desarrollo de software tradicional, que se basa en una secuencia de fases lineales, donde cada fase se completa antes de pasar a la siguiente. Es decir, es una metodología que se enfoca en la planificación y el seguimiento de un camino predeterminado, sin permitir cambios ni retroalimentación durante el proceso. En resumen, la principal característica de la metodología Cascada no se encuentra en los documentos proporcionados, pero se puede describir como una metodología que se enfoca en la planificación y el seguimiento de un camino predeterminado, sin permitir cambios ni retroalimentación durante el proceso.

### Respuesta Esperada:

La principal característica que define a la metodología de desarrollo de software conocida como Cascada es su enfoque tradicional y secuencial.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Ejemplo: Metodologías Tradicionales RUP, MSF, .... Metodologías Ágiles SCRUM, XP, Kanban .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

2. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....

3. metodología ágil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado.....

### Métricas de Evaluación:

Answer Relevancy: 0.78 (Threshold: 0.5) - The score is 0.78 because although the output provides some relevant information, it includes redundant and unimportant details that don't directly address the principal characteristic of the Cascada methodology.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the first node which talks about agile methodologies, the second node which explains software engineering and agile methodologies, and the third node which explains Scrum methodology, are irrelevant to the input and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context does not mention the Cascada methodology at all, as stated in the reasons for irrelevancy, such as 'The statement does not mention the Cascada methodology.'



## Caso de Prueba: 80 - Fallo

### Input del Usuario:

¿Cuáles son las fases del ciclo de vida del Rational Unified Process (RUP)?

### Respuesta Generada:

Aunque el contexto proporcionado no menciona explícitamente el Rational Unified Process (RUP), podemos inferir las fases del ciclo de vida del desarrollo de software a partir de los documentos proporcionados. Aunque no hay una respuesta directa en los documentos, podemos identificar las fases del ciclo de vida del desarrollo de software a partir de los documentos 1, 3, 5 y 8. Las fases del ciclo de vida del desarrollo de software, que coinciden con las fases del Rational Unified Process (RUP), son: 1. **\*\*Inception\*\*** (Concepción): Fase de ideación y definición del problema a resolver, donde se interactúa con los stakeholders y usuarios para entender los procesos y recopilar los requisitos del proyecto. 2. **\*\*Elaboration\*\*** (Elaboración): Fase de análisis y especificación de los requisitos funcionales y no funcionales del sistema, donde se define la arquitectura del sistema de software. 3. **\*\*Construction\*\*** (Construcción): Fase de desarrollo del sistema de software, utilizando una metodología de desarrollo y lenguajes de programación. 4. **\*\*Transition\*\*** (Transición): Fase de evaluación de la calidad del sistema, pruebas y despliegue del sistema en los servidores o ambiente donde va a funcionar. Además, se mencionan otras fases en los documentos, como el monitoreo del sistema una vez que está en producción. Es importante destacar que, aunque no se menciona explícitamente el RUP, las fases identificadas coinciden con las fases del proceso de desarrollo de software que se utilizan comúnmente en la industria.

### Respuesta Esperada:

No se puede responder a la pregunta con el contexto proporcionado, ya que no se mencionan las fases del ciclo de vida del Rational Unified Process (RUP) en el contexto. El contexto solo menciona que RUP es una metodología tradicional de desarrollo de software, pero no proporciona información sobre sus fases.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Ejemplo: Metodologías Tradicionales RUP, MSF, .... Metodologías Ágiles SCRUM, XP, Kanban .....

### Contexto Recuperado:

1. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy día interactuar con los stake.....

2. conceptos lo novedoso agil devops ia automatizacion de partes del ciclo de vida despliegue pruebas y parte de la codificacion automatizacion de todas las etapas agil ingenieria de software aumentad.....

3. conceptos metodologias de desarrollo de software una metodologia de desarrollo de software es un conjunto estructurado de practicas principios y procedimientos que guian y regulan el proceso de creaci.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.91 (Threshold: 0.5) - The score is 0.91 because the response is mostly relevant, but slightly loses focus by including a commentary on the lack of explicit mention of RUP, instead of solely focusing on the phases of RUP.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant, with the first node failing to mention specific RUP phases, the second node discussing agile, devops, and automation instead of RUP, and the third node only talking about software development methodologies in general.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context, as the context only provides general information about RUP being a traditional software development methodology, but lacks specific details.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context is only partially relevant, as it mentions 'ciclo de vida del desarrollo de software' and other related concepts, but does not specifically provide the phases of the Rational Unified Process (RUP) lifecycle, as stated in the input.

## Caso de Prueba: 81 - Fallo

### Input del Usuario:

¿Cuáles son los pilares fundamentales de la metodología de desarrollo de software MSF (Microsoft Solutions Framework)?

### Respuesta Generada:

Basándome en los documentos proporcionados, no se menciona explícitamente la metodología de desarrollo de software MSF (Microsoft Solutions Framework) en ninguno de ellos. Los documentos se centran en conceptos generales de la ingeniería de software, metodologías ágiles de desarrollo de software como Scrum, y otros aspectos relacionados con el desarrollo de software. Sin embargo, puedo proporcionar información sobre los pilares fundamentales de la metodología de desarrollo de software MSF (Microsoft Solutions Framework) en general. Los pilares fundamentales de MSF son: 1. **Team Model**: Define los roles y responsabilidades del equipo de desarrollo, y cómo trabajar juntos para alcanzar los objetivos del proyecto. 2. **Process Model**: Describe las fases y actividades necesarias para desarrollar un proyecto de software, incluyendo la planificación, el desarrollo, la verificación y la implementación. 3. **Discipline**: Establece las prácticas y procedimientos para garantizar la calidad y la consistencia en el desarrollo del software. 4. **Risk Management**: Identifica y gestiona los riesgos del proyecto para minimizar su impacto en el desarrollo del software. Es importante destacar que, como no se menciona la metodología MSF en los documentos proporcionados, esta respuesta se basa en conocimientos generales sobre la metodología MSF.

### Respuesta Esperada:

Según el contexto proporcionado, no se mencionan los pilares fundamentales de la metodología de desarrollo de software MSF (Microsoft Solutions Framework). Solo se menciona que MSF es un ejemplo de metodología tradicional de desarrollo de software, pero no se proporciona información adicional sobre sus pilares fundamentales.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Ejemplo: Metodologías Tradicionales RUP, MSF, .... Metodologías Ágiles SCRUM, XP, Kanban .....

### Contexto Recuperado:

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....
2. metodología ágil de desarrollo de software scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado.....
3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because while the output provides some relevant information, it is brought down by the presence of multiple irrelevant statements that fail to address the fundamental pillars of MSF.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant to the input, with the first node failing to mention the fundamental pillars of MSF, the second node discussing Scrum instead of MSF, and the third node being a repeated text that only mentions MSF as an example.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the output only partially matches the retrieval context, specifically sentence 1 mentioning 'metodología tradicional de desarrollo de software' which is related to nodes 1 and 3 in the retrieval context, but lacks information about the fundamental pillars of MSF.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely irrelevant to the input, as all statements mention unrelated topics such as 'objetivos del curso', agile methodologies, Scrum, and traditional software development methodologies, but none of them mention MSF methodology.

## Caso de Prueba: 82 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre las metodologías ágiles y las metodologías tradicionales en el desarrollo de software?

### Respuesta Generada:

Según el Documento 4, las metodologías ágiles y tradicionales son dos enfoques diferentes en el desarrollo de software. Las metodologías ágiles se centran en la flexibilidad, la iteración y la entrega continua, mientras que las metodologías tradicionales se centran en la planificación, la predicción y la ejecución de un plan establecido. La principal diferencia entre las metodologías ágiles y tradicionales es la forma en que abordan el cambio y la incertidumbre en el desarrollo de software. Las metodologías ágiles asumen que el requisito cambia y que es necesario adaptarse a estos cambios rápidamente, mientras que las metodologías tradicionales intentan predecir y planificar todos los requisitos desde el principio. En el Documento 1, se mencionan los 12 principios del Manifiesto Ágil, que incluyen la atención continua a la excelencia técnica y al buen diseño, la mejora de la agilidad y la simplicidad, y el arte de maximizar la cantidad de trabajo realizado. Esto se logra a través de la iteración y la reflexión continua, lo que permite a los equipos adaptarse a los cambios y mejorar su proceso de desarrollo. Por otro lado, las metodologías tradicionales se centran en la planificación y la ejecución de un plan establecido, lo que puede llevar a una mayor rigidez y menor capacidad de adaptación a los cambios. En resumen, la principal diferencia entre las metodologías ágiles y tradicionales es la forma en que abordan el cambio y la incertidumbre, con las metodologías ágiles siendo más flexibles y adaptativas, y las metodologías tradicionales siendo más planificadas y rígidas.

### Respuesta Esperada:

La principal diferencia entre las metodologías ágiles y las metodologías tradicionales en el desarrollo de software es que las primeras, como SCRUM, XP y Kanban, se centran en la flexibilidad y la adaptabilidad, mientras que las segundas, como RUP y MSF, se basan en un enfoque más rígido y predeterminado.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Ejemplo: Metodologías Tradicionales RUP, MSF, .... Metodologías Ágiles SCRUM, XP, Kanban .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

2. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

3. clase anterior taller conceptos basicos del desarrollo de una aplicacion con django estimacion de costos y tiempo requerido para el desarrollo de una aplicacion usando metodologias agiles conceptos

ba.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.79 (Threshold: 0.5) - The score is 0.79 because the response provided some relevant information, but was distracted by the unnecessary explanation of the 12 principles of the Agile Manifesto, which didn't directly address the main question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 discussing 'metodologias agiles' and node 2 mentioning 'entender y usar metodologias agiles de desarrollo de software', are ranked higher than the irrelevant node 3, which is about 'clase anterior taller conceptos basicos del desarrollo de una aplicacion con django', showing a perfect ranking of relevant nodes over irrelevant ones.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output lacks any connection to nodes in retrieval context.

Contextual Relevancy: 0.61 (Threshold: 0.5) - The score is 0.61 because the retrieval context has many irrelevant statements about course objectives, software engineering concepts, and roles in software development, as stated in the reasons for irrelevancy, but still contains some relevant statements about agile methodologies, such as 'metodologias agiles 12 principios definidos en el manifiesto agil' and 'entender y usar metodologias agiles de desarrollo de software'.

## Caso de Prueba: 83 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal de la metodología Kanban en el desarrollo de software?

### Respuesta Generada:

Basándome en la información proporcionada en los documentos, no encontré una respuesta directa a la pregunta sobre el objetivo principal de la metodología Kanban en el desarrollo de software. Sin embargo, puedo proporcionar contexto y información relacionada que puede ayudar a responder a la pregunta. Los documentos mencionan diferentes aspectos del desarrollo de software ágil, como la importancia de tener software funcionando, la atención continua a la excelencia técnica y la simplicidad, la emergencia de las mejores arquitecturas y diseño, y la importancia de la iteratividad y la colaboración en equipo. Sin embargo, no se menciona explícitamente la metodología Kanban. Kanban es una metodología ágil que se enfoca en la visualización del flujo de trabajo, la limitación del trabajo en progreso y la mejora continua. El objetivo principal de Kanban es maximizar la eficiencia del flujo de trabajo y la entrega de valor al cliente, logrando eso a través de la visualización, la limitación y la mejora continua. En general, el objetivo principal de Kanban en el desarrollo de software es maximizar la entrega de valor al cliente, reducir el tiempo de entrega y mejorar la calidad del software, mediante la visualización y la gestión del flujo de trabajo y la limitación del trabajo en progreso.

### Respuesta Esperada:

No se puede determinar el objetivo principal de la metodología Kanban en el desarrollo de software solo con el contexto proporcionado, ya que no se ofrece información específica sobre Kanban. Sin embargo, se puede inferir que Kanban es una metodología ágil, pero no se puede determinar su objetivo principal con certeza.

### Contexto Relevante:

1. Conceptos Metodologías de desarrollo de software Ejemplo: Metodologías Tradicionales RUP, MSF, .... Metodologías Ágiles SCRUM, XP, Kanban .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

2. desarrollo de software ágil el desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles se concibieron como una alternativa a las pract.....

3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the response partially addresses the question, but lacks specificity and provides alternative objectives, making it only half relevant.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant to the input, with the first node not mentioning Kanban, the second node only discussing agile development, and the third node focusing on software engineering and agile methodologies in general.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context does not mention Kanban methodology at all, as stated in multiple reasons for irrelevancy, such as 'The statement does not mention Kanban methodology.' and 'The statement talks about agile processes in general, not specifically about Kanban methodology.'



## Caso de Prueba: 84 - Éxito

### Input del Usuario:

¿Cuál es el papel del legado en la implementación de metodologías tradicionales en la gestión de proyectos?

### Respuesta Generada:

Según el Documento 2, el concepto "legado" se refiere a las metodologías tradicionales en el desarrollo de software. En este sentido, el papel del legado en la implementación de metodologías tradicionales en la gestión de proyectos se refiere al uso de enfoques y técnicas establecidos para gestionar y desarrollar proyectos de software. En este contexto, el legado se enfoca en la definición y seguimiento de requerimientos funcionales, como se muestra en el Documento 1, donde se presentan ejemplos de requerimientos funcionales en metodologías tradicionales, tales como crear un usuario del sistema, registrar participantes a un evento, registrar la venta de un artículo, anular una nota débito, entre otros. Además, el legado implica la utilización de artefactos y técnicas específicas, como la análisis y especificación de requerimientos funcionales y no funcionales, como se menciona en el Documento 8. Esto incluye la creación de un producto backlog, la planificación y el desarrollo de un producto de software, y la gestión de proyectos a través de metodologías tradicionales como RUP y MSF. En resumen, el papel del legado en la implementación de metodologías tradicionales en la gestión de proyectos se enfoca en la aplicación de enfoques y técnicas establecidos para definir y gestionar los requerimientos del proyecto, y desarrollar el software de manera estructurada y planificada.

### Respuesta Esperada:

El papel del legado en la implementación de metodologías tradicionales en la gestión de proyectos es el de obstaculizar o frenar el cambio hacia nuevas formas de trabajar. El legado se refiere a las prácticas, procesos y sistemas heredados que se han utilizado en el pasado y que pueden no ser compatibles con las nuevas metodologías. En este sentido, el legado puede ser un obstáculo para la implementación de nuevas prácticas y procesos, ya que los miembros del equipo pueden ser reacios a cambiar sus formas de trabajar habituales.

### Contexto Relevante:

1. Conceptos: El legado Metodologías tradicionales...

### Contexto Recuperado:

1. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....
2. conceptos el legado metodologías tradicionales...
3. requerimientos funcionales requerimientos en las metodologías tradicionales...

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the answer is highly relevant to the input, but slightly loses focus by mentioning Agile methodologies, which are not directly related to traditional methodologies.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first and third nodes in the retrieval contexts, which are not relevant to the input, are ranked equally with the second node, which is directly related to the topic and provides a clear connection between legacy and traditional methodologies.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, as the 2nd node in the retrieval context explicitly mentions key concepts like 'el legado' and 'metodologias tradicionales', allowing for a seamless connection.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context is perfectly relevant to the input, with statements like 'conceptos el legado metodologias tradicionales' directly addressing the role of legacy in traditional methodologies, making it a perfect match!

## Caso de Prueba: 85 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre la metodología Waterfall y la metodología V-Model en términos de enfoque y ciclo de vida del proyecto?

### Respuesta Generada:

La principal diferencia entre la metodología Waterfall y la metodología V-Model se encuentra en su enfoque y ciclo de vida del proyecto. La metodología Waterfall es una metodología tradicional que sigue un enfoque secuencial y lineal. El ciclo de vida del proyecto se divide en fases discretas y separadas, que se ejecutan de manera secuencial, sin retroalimentación ni iteración entre ellas. Cada fase debe completarse antes de pasar a la siguiente, lo que implica que no hay vuelta atrás en el proceso de desarrollo. Este enfoque es riesgoso, ya que no permite cambios en el proyecto una vez que se ha iniciado. Por otro lado, la metodología V-Model es también una metodología tradicional, pero con un enfoque más flexible y cíclico. El ciclo de vida del proyecto se presenta en forma de "V", con fases de análisis, diseño, implementación, pruebas y mantenimiento. Cada fase se relaciona con la fase correspondiente en la otra parte del "V", lo que permite una retroalimentación y iteración continua entre ellas. Esto permite identificar y corregir errores y cambios en el proyecto de manera más eficiente. En resumen, la principal diferencia entre Waterfall y V-Model es que Waterfall es secuencial y no permite cambios, mientras que V-Model es cíclico y permite iteración y retroalimentación entre fases. Es importante destacar que, según el Documento 6, las metodologías de desarrollo de software establecen un marco para organizar y gestionar las tareas involucradas en cada fase del ciclo de vida del software, y que tanto Waterfall como V-Model son consideradas metodologías tradicionales.

### Respuesta Esperada:

La principal diferencia entre la metodología Waterfall y la metodología V-Model radica en su enfoque y ciclo de vida del proyecto. La metodología Waterfall sigue un enfoque secuencial y lineal, donde cada fase se completa antes de pasar a la siguiente, sin retroalimentación ni iteraciones. Por otro lado, la metodología V-Model tiene un enfoque en forma de "V", donde la fase de desarrollo se cruza con la fase de testing, permitiendo una retroalimentación y iteración continua entre ambas.

### Contexto Relevante:

1. Conceptos: El legado Metodologías tradicionales...

### Contexto Recuperado:

1. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....
2. tareas investigar la diferencia entre get y post get 1 función el método get se utiliza para solicitar datos de un servidor es decir se usa para recuperar información por ejemplo una página web o los .....
3. presentación de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escritas.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and relevant comparison between Waterfall and V-Model methodologies!

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant to the input, with the first node not mentioning Waterfall or V-Model methodologies, the second node explaining HTTP requests, and the third node providing software engineering definitions, none of which address the input question.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context is completely irrelevant to the input, as all the statements are about course objectives, software engineering concepts, agile development, HTTP methods, and other unrelated topics, as stated in the reasons for irrelevancy.

## Caso de Prueba: 86 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de las metodologías ágiles en el desarrollo de software?

### Respuesta Generada:

Según el Documento 1, el propósito principal de las metodologías ágiles en el desarrollo de software es promover el desarrollo sostenible, es decir, enfocarse en la entrega de software funcionando de manera continua y sostenible en el tiempo. Esto se logra a través de la atención continua a la excelencia técnica y al buen diseño, lo que a su vez mejora la agilidad y la simplicidad en el desarrollo de software. Además, según el Documento 2, las metodologías ágiles buscan ofrecer una forma diferente de gestionar equipos y proyectos de desarrollo de software, alejándose de las prácticas tradicionales y enfocándose en la entrega de valor al cliente de manera rápida y flexible. En resumen, el propósito principal de las metodologías ágiles es entregar software funcionando de manera continua y sostenible, mediante la atención a la excelencia técnica y el buen diseño, y ofreciendo una forma diferente de gestionar equipos y proyectos de desarrollo de software.

### Respuesta Esperada:

El propósito principal de las metodologías ágiles en el desarrollo de software es basarse en el desarrollo iterativo e incremental.

### Contexto Relevante:

1. Conceptos: Metodologías actuales Metodologías ágiles Las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Las metodologías ágiles .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question about the main purpose of agile methodologies in software development, providing a clear and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which mentions 'el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles' and node 2 which highlights 'el desarrollo agil es una forma diferente de gestionar equipos y proyectos de desarrollo de software', are ranked higher than the irrelevant node 3 which talks about 'objetivos del curso'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in retrieval context, which explicitly discusses agile methodologies in software development.

Contextual Relevancy: 0.69 (Threshold: 0.5) - The score is 0.69 because the retrieval context has some relevant statements that align with the input, such as 'el software funcionando es la medida principal de progreso' and 'metodologias agiles 12 principios definidos en el manifiesto agil', but is brought down by irrelevant statements that talk about course objectives, traditional software development methodologies, and understanding roles and activities, as mentioned in the reasons for irrelevancy.

## Caso de Prueba: 87 - Éxito

### Input del Usuario:

¿Cuál es el objetivo principal de las metodologías ágiles en el desarrollo de software?

### Respuesta Generada:

Según el Documento 1, el objetivo principal de las metodologías ágiles en el desarrollo de software es tener "el software funcionando", es decir, la medida principal de progreso es que el software esté funcionando correctamente. Además, se enfatiza la importancia de la atención continua a la excelencia técnica y el buen diseño para mejorar la agilidad y la simplicidad. En otras palabras, el objetivo principal de las metodologías ágiles es entregar software que funcione correctamente y que sea sostenible, flexible y adaptable a los cambios, mediante el trabajo en equipo, la colaboración y la reflexión continua sobre cómo ser más efectivos. Es importante destacar que, según el Documento 8, el desarrollo ágil de software se basa en 4 valores fundamentales: individuals and interactions (individuos e interacciones), working software (software funcionando), customer collaboration (colaboración con el cliente) y responding to change (respuesta al cambio).

### Respuesta Esperada:

El objetivo principal de las metodologías ágiles en el desarrollo de software es el desarrollo iterativo e incremental.

### Contexto Relevante:

1. Conceptos: Metodologías actuales Metodologías ágiles Las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Las metodologías ágiles .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as node 1 which mentions iterative and incremental development, and node 2 which explains the concept of agile development, are ranked higher than the irrelevant node 3 which discusses course objectives and traditional software development methodologies.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in retrieval context, which explicitly mentions agile methodologies in software development.

Contextual Relevancy: 0.69 (Threshold: 0.5) - The score is 0.69 because the retrieval context contains some relevant statements about agile methodologies, such as 'el software funcionando es la medida principal de progreso' and 'metodologías ágiles 12 principios definidos en el manifiesto ágil', but is largely irrelevant due to the presence of unrelated terms and statements about course objectives, software engineering concepts, and traditional software development methodologies, as mentioned in the reasons for irrelevancy.



## Caso de Prueba: 88 - Éxito

### Input del Usuario:

¿Cuál es el enfoque principal de las metodologías ágiles en el desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, el enfoque principal de las metodologías ágiles en el desarrollo de software es la entrega incremental y colaborativa de software funcionando, enfatizando la atención continua a la excelencia técnica y al buen diseño. Esto se logra a través de procesos iterativos que promueven la entrega de pequeñas partes del software en cada iteración, permitiendo la retroalimentación continua y el ajuste de los procesos para ser más efectivos. Además, las metodologías ágiles se centran en la colaboración en equipo, la planificación continua y el aprendizaje continuo, lo que permite responder de manera efectiva a los cambios y a las necesidades del cliente. En lugar de intentar entregar todo el software de una vez al final del proyecto, las metodologías ágiles se enfocan en entregar pequeñas partes del software funcionando en cada iteración.

### Respuesta Esperada:

El enfoque principal de las metodologías ágiles en el desarrollo de software es el desarrollo iterativo e incremental.

### Contexto Relevante:

1. Conceptos: Metodologías actuales Metodologías ágiles Las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Las metodologías ágiles .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a direct and accurate response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which mentions 'el desarrollo sostenible' and node 2 which mentions 'el desarrollo agil de software', are ranked higher than the irrelevant node 3 which talks about 'requerimientos en las metodologias agiles' and does not address the main focus of agile methodologies.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in retrieval context, which explicitly mentions 'desarrollo de software agil' and 'metodologias agiles', aligning with the expected output.

Contextual Relevancy: 0.73 (Threshold: 0.5) - The score is 0.73 because the retrieval context has some relevant statements that align with the input, such as 'el software funcionando es la medida principal de progreso' and 'metodologias agiles 12 principios definidos en el manifiesto agil', but is also cluttered with irrelevant information, like 'scrumcolombia desarrollo de software i' and statements focused on user stories implementation, which dilutes the overall relevance.

## Caso de Prueba: 89 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal del desarrollo incremental en las metodologías ágiles?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal del desarrollo incremental en las metodologías ágiles es la entrega incremental del software, enfatizando la colaboración en equipo, la planificación continua y el aprendizaje continuo. En lugar de intentar entregar todo de una vez al final del proyecto, se entrega el software en pequeñas partes funcionales, permitiendo así una retroalimentación temprana y la adaptación a los cambios en los requisitos del usuario. En otras palabras, el desarrollo incremental permite: \* Entregar valor al usuario de manera regular \* Recibir retroalimentación temprana y adaptarse a los cambios \* Reducir el riesgo de no cumplir con los requisitos del usuario \* Mejorar la calidad del software mediante la iteración y la reflexión continua En resumen, el desarrollo incremental es un enfoque clave en las metodologías ágiles, que busca entregar software de manera iterativa e incremental, con el fin de mejorar la calidad, reducir el riesgo y aumentar la satisfacción del usuario.

### Respuesta Esperada:

El propósito principal del desarrollo incremental en las metodologías ágiles es basar el desarrollo de software en un proceso iterativo e incremental.

### Contexto Relevante:

1. Conceptos: Metodologías actuales Metodologías ágiles Las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Las metodologías ágiles .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

2. desarrollo de software ágil el desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles se concibieron como una alternativa a las pract.....

3. conceptos metodologías actuales metodologías ágiles las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental las metodologías ágiles se.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts directly addresses the question, with the top two nodes being irrelevant. For instance, the first node at rank 1 is about software being functional, and the second node at rank 2 explains agile development, but neither of them answers the question about the main purpose of incremental development. The relevant node at rank 3 finally provides the correct information.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output sentence perfectly aligns with the 3rd node in the retrieval context, which also emphasizes iterative and incremental software development.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context contains some relevant statements about agile methodologies, such as 'el software funcionando es la medida principal de progreso' and 'el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles', but is largely irrelevant due to the presence of unrelated information and incomplete statements.

## Caso de Prueba: 90 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de las prácticas ágiles en el desarrollo de software?

### Respuesta Generada:

Según el contexto proporcionado, el propósito principal de las prácticas ágiles en el desarrollo de software es promover el desarrollo sostenible, la atención continua a la excelencia técnica y al buen diseño, y maximizar la cantidad de trabajo realizado. Esto se logra a través de la reflexión regular del equipo sobre cómo ser más efectivo y ajustar su comportamiento en consecuencia. En otras palabras, las prácticas ágiles se enfocan en la entrega de software funcionando y de alta calidad, con una mentalidad de colaboración y respuesta al cambio, y con una atención continua a la excelencia técnica y al buen diseño. Esto se resume en los 4 valores y 12 principios del Manifiesto Ágil, que definen la esencia de la filosofía ágil. En este sentido, las prácticas ágiles buscan mejorar la agilidad y la simplicidad en el desarrollo de software, permitiendo a los equipos autoorganizados y a los individuos trabajar de manera más efectiva y eficiente, y entregando software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

El propósito principal de las prácticas ágiles en el desarrollo de software es implementar el desarrollo iterativo e incremental.

### Contexto Relevante:

1. Conceptos: Metodologías actuales Metodologías ágiles Las metodologías ágiles se refieren a métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Las metodologías ágiles .....

### Contexto Recuperado:

1. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

3. como iniciar el desarrollo de proyecto de software usando practicas agiles...

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is highly relevant to the input, but slightly loses focus by mentioning a secondary aspect of agile practices.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which only discusses starting a project with agile practices but doesn't state the main purpose.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the sentence perfectly matches the 2nd node in the retrieval context, which also discusses agile practices in software development.

Contextual Relevancy: 0.82 (Threshold: 0.5) - The score is 0.82 because the retrieval context, although containing some irrelevant information like 'metodologias agiles 12 principios definidos en el manifiesto agil' and 'scrumcolombia desarrollo de software i', still provides several relevant statements that align with the main purpose of agile practices in software development, such as 'el software funcionando es la medida principal de progreso' and 'el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles'.

## Caso de Prueba: 91 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre Scrum y Kanban en un proyecto de desarrollo de software?

### Respuesta Generada:

La principal diferencia entre Scrum y Kanban en un proyecto de desarrollo de software es que Scrum es una metodología ágil que implica la creación de equipos de trabajo, llamados "Scrum teams", que se enfocan en la entrega de incrementos de valor al final de cada sprint (un período de tiempo determinado, generalmente de 2-4 semanas). Estos equipos trabajan de manera iterativa, priorizando tareas y realizando reuniones diarias para sincronizar y hacer seguimiento al trabajo. Por otro lado, Kanban es una metodología ágil que se centra en la visualización del flujo de trabajo, la limitación del trabajo en proceso y la entrega continua de valor. Kanban no impone la creación de equipos ni la estructura de sprint, sino que se enfoca en la gestión del flujo de trabajo y la entrega de valor de manera continua. En resumen, Scrum es una metodología ágil más estructurada que implica la creación de equipos y sprint, mientras que Kanban es una metodología ágil más flexible que se centra en la visualización y gestión del flujo de trabajo. Es importante destacar que aunque ambos enfoques sean ágiles, Scrum es una metodología más prescriptiva que Kanban, y Kanban es más adaptable a las necesidades específicas del proyecto y del equipo.

### Respuesta Esperada:

La principal diferencia entre Scrum y Kanban en un proyecto de desarrollo de software es que Scrum tiene un enfoque en iteraciones o sprints, mientras que Kanban no tiene iteraciones ni sprints, sino que se centra en el flujo continuo de trabajo.

### Contexto Relevante:

1. Conceptos Metodologías ágiles Metodologías ágiles Las más usadas Scrum XP Kanban Scrum + XP + Kanban...

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....
2. pregunta en conferencia se reunen los representantes de los equipos por cada nivel nivel 1 5 veces nivel 2 2 veces y nivel3 1 vez por semana duracion 15 30 y 45 minutos respectivamente preguntas durante.....
3. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the response partially addresses the question by explaining Scrum, but falls short of fully comparing it to Kanban, leaving a significant aspect of the question unanswered.

Contextual Precision: 0.58 (Threshold: 0.5) - The score is 0.58 because the first node in the retrieval context, which is irrelevant to the input, is ranked higher than the second and third nodes, which provide valuable insights into Scrum methodology and mention both Scrum and Kanban, respectively. For instance, the first node's reason is that it 'does not provide any information about the differences between Scrum and Kanban', whereas the second node mentions 'un tamaño adecuado que se recomienda para un equipo de scrum' and the third node mentions 'metodologías ágiles para el desarrollo de software scrum xp kanban y otras'.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.24 (Threshold: 0.5) - The score is 0.24 because the retrieval context mostly contains irrelevant statements that do not mention Scrum or Kanban, or discuss unrelated topics, as stated in the reasons for irrelevancy, such as 'The statement does not mention Scrum or Kanban...' or 'The statement is completely unrelated to Scrum or Kanban, it talks about web development concepts.'



## Caso de Prueba: 92 - Fallo

### Input del Usuario:

¿Cuál es el papel del Product Owner en Scrum?

### Respuesta Generada:

Basándome en la información proporcionada en los documentos, no se encuentra una respuesta directa al papel del Product Owner en Scrum. Sin embargo, puedo inferir que el Product Owner es el responsable del Product Backlog, que es una lista priorizada y estimada de las historias de usuario que representan las funcionalidades del sistema que se va a construir. En este sentido, el papel del Product Owner sería: \* Gestionar y priorizar el Product Backlog, asegurándose de que las historias de usuario estén claramente definidas y priorizadas según las necesidades del cliente y del proyecto. \* Asegurarse de que el equipo de desarrollo tenga una comprensión clara de las historias de usuario y los requisitos del sistema. \* Trabajar con el equipo de desarrollo para asegurarse de que las historias de usuario se implementen de acuerdo con los requisitos y expectativas del cliente. \* Asegurarse de que el incremento de producto sea lo que el cliente espera y necesita. Aunque no se menciona explícitamente el papel del Product Owner en los documentos, es común que en Scrum el Product Owner sea el responsable del Product Backlog y tenga una función clave en la gestión de los requisitos y la priorización de las historias de usuario.

### Respuesta Esperada:

No hay suficiente información en el contexto proporcionado para responder a la pregunta. El contexto solo menciona los nombres de algunas metodologías ágiles, pero no proporciona información sobre el papel del Product Owner en Scrum.

### Contexto Relevante:

1. Conceptos Metodologías ágiles Metodologías ágiles Las más usadas Scrum XP Kanban Scrum + XP + Kanban...

### Contexto Recuperado:

1. practicas agiles en scrum definition of done el definition of done es un conjunto de criterios objetivos que definen cuando una historia de usuario una tarea o un incremento del producto se co.....
2. practicas agiles en scrum product backlog una lista priorizada y estimada de las historias de usuario que representan las funcionalidades del sistema que se va a construir en algunas ocasiones se inc.....
3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

### Métricas de Evaluación:

Answer Relevancy: 0.82 (Threshold: 0.5) - The score is 0.82 because the response is mostly relevant, but slightly derailed by a negative response and a statement about the absence of information, which

didn't fully address the role of the Product Owner in Scrum.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts are irrelevant, with the first node not mentioning the role of the Product Owner, the second node only discussing the product backlog, and the third node focusing on the retrospective meeting.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output (sentence 1) can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.15 (Threshold: 0.5) - The score is 0.15 because the retrieval context barely mentions Scrum and related terms like 'practicas agiles en scrum' and 'product backlog', but fails to provide any substantial information about the role of the Product Owner, as stated in the reasons for irrelevancy, such as 'The statement does not mention the Product Owner's role.'

## Caso de Prueba: 93 - Fallo

### Input del Usuario:

¿Cuál es el enfoque principal de la metodología ágil XP en cuanto a la programación en parejas?

### Respuesta Generada:

Primero, debemos aclarar que la pregunta se refiere a la metodología ágil XP (eXtreme Programming), pero no se encuentra una mención explícita a XP en los documentos proporcionados. Sin embargo, podemos inferir la respuesta basándonos en los conceptos y enfoques generales de las metodologías ágiles presentados en los documentos. En cuanto a la programación en parejas, no se proporciona información específica en los documentos. Sin embargo, se pueden deducir algunas conclusiones. La programación en parejas es una práctica común en varias metodologías ágiles, incluyendo XP. El enfoque principal de esta práctica es la colaboración en equipo y el aprendizaje continuo. De acuerdo con el Documento 1, el desarrollo de software ágil enfatiza la entrega incremental, la colaboración en equipo y el aprendizaje continuo. La programación en parejas se ajusta a este enfoque, ya que dos desarrolladores trabajan juntos en la misma tarea, intercambian ideas y habilidades, y se apoyan mutuamente. En conclusión, aunque no se menciona explícitamente la programación en parejas en los documentos, se puede inferir que el enfoque principal es la colaboración en equipo y el aprendizaje continuo, que es un aspecto fundamental de las metodologías ágiles.

### Respuesta Esperada:

El enfoque principal de la metodología ágil XP en cuanto a la programación en parejas es la programación en pareja.

### Contexto Relevante:

1. Conceptos Metodologías ágiles Metodologías ágiles Las más usadas Scrum XP Kanban Scrum + XP + Kanban...

### Contexto Recuperado:

1. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. estimar historias de usuario practicas agiles en scrum la estimacion en la metodologia agil consiste en asignar puntos a una tarea o historia de usuario no hay una formula para realizarlo de una maner.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the output is mostly relevant to the input, but has some minor distractions that prevent it from being a perfect match.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are irrelevant to the input, with the first two nodes not mentioning 'programación en parejas' at all, and the third node only discussing estimation in agile methodology without any mention of 'programación en parejas'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, as the 1st node in the retrieval context directly mentions 'agil', which is a key concept in the sentence about 'metodología ágil XP'.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context does not mention 'programación en parejas' or XP methodology, and only talks about agile development in general, making it completely irrelevant to the input.

## Caso de Prueba: 94 - Fallo

### Input del Usuario:

¿Cuál es el principal objetivo de la columna "Done" en una tabla de Kanban?

### Respuesta Generada:

Según el Documento 2, el principal objetivo de la columna "Done" (también conocida como Definition of Done, DoD) es asegurar la calidad, coherencia y alineación del equipo en cuanto a las expectativas de finalización. El DoD define cuando una historia de usuario, tarea o incremento del producto se considera completado en un proyecto ágil.

### Respuesta Esperada:

Basándome en el contexto proporcionado, no hay suficiente información para responder a la pregunta. El contexto solo menciona los nombres de las metodologías ágiles, pero no proporciona detalles sobre la columna "Done" en una tabla de Kanban.

### Contexto Relevante:

1. Conceptos Metodologías ágiles Metodologías ágiles Las más usadas Scrum XP Kanban Scrum + XP + Kanban...

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....
2. practicas agiles en scrum dod definition of done el definition of done dod es un conjunto de criterios objetivos que definen cuando una historia de usuario una tarea o un incremento del producto se co.....
3. introduccion a bpm bpm business process model y bpmn business process model es un estandar de modelado desarrollado por la object management group omg que proporciona un lenguaje grafico para document.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the node ranked 2, which mentions the 'Definition of Done' (DoD) related to the 'Done' column in a Kanban table, is correctly ranked higher than the irrelevant nodes, such as node ranked 1, which talks about project objectives and planning, and node ranked 3, which discusses Business Process Model and BPMN, but the irrelevant nodes are not ranked low enough to achieve a higher score.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output relate to any node in the retrieval context.

Contextual Relevancy: 0.27 (Threshold: 0.5) - The score is 0.27 because the retrieval context mostly contains irrelevant information, with multiple statements stating 'The statement is not related to the 'Done' column in a Kanban table', and only a few relevant statements, such as 'el definition of done dod es un conjunto de criterios objetivos que definen cuando una historia de usuario una tarea o un incremento del producto se considera completado en un proyecto agil', which are not enough to significantly impact the score.

## Caso de Prueba: 95 - Éxito

### Input del Usuario:

¿Cuáles son las características clave que debe tener un requisito de software según la definición?

### Respuesta Generada:

Según el Documento 1 y el Documento 6, las características clave que debe tener un requisito de software son: 1. **\*\*Necesario\*\***: Lo que pida un requisito debe ser necesario para el producto correcto. 2. **\*\*No ambiguo\*\***: El texto debe ser claro, preciso y tener una única interpretación posible. 3. **\*\*Conciso\*\***: Debe redactarse en un lenguaje comprensible por los clientes, en lugar de uno de tipo técnico y especializado. 4. **\*\*Consistente\*\***: Ningún requisito debe entrar en conflicto con otro requisito diferente, y el lenguaje empleado entre los distintos requisitos debe ser consistente. 5. **\*\*Completo\*\***: Los requisitos deben contener en sí mismos toda la información necesaria y no remitir a otras fuentes externas que los expliquen con más detalle. 6. **\*\*Alcanzable\*\***: Un requisito debe ser un objetivo realista posible de ser alcanzado en el tiempo y los recursos disponibles. 7. **\*\*Verificable\*\***: Se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no, lo que puede lograrse mediante inspección, análisis, demostración o testeo. Es importante destacar que estas características son fundamentales para garantizar que los requisitos de software sean claros, precisos y alcanzables, lo que a su vez permitirá desarrollar un software que cumpla con las necesidades y expectativas de los usuarios y clientes.

### Respuesta Esperada:

Según la definición, las características clave que debe tener un requisito de software son: \* No ambiguo \* Comprobable o medible \* Necesario para la aceptación del producto o proceso.

### Contexto Relevante:

1. REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitación es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación.....
3. detalles de una historia de usuario criterios de aceptación ítem criterio de aceptación las condiciones que un producto de software debe satisfacer para ser aceptado por un usuario o cliente un usuario.....

### Métricas de Evaluación:

Answer Relevancy: 0.89 (Threshold: 0.5) - The score is 0.89 because the output is highly relevant, but slightly deducted for explaining the importance of characteristics instead of directly listing them.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like the first node that mentions 'no ambiguo', 'comprobable o medible', and 'necesario para la aceptación del producto o proceso', are ranked higher than the irrelevant nodes, like the second and third nodes, which are about the process of specifying requirements and user stories and acceptance criteria respectively.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 1st node in the retrieval context, showcasing a seamless connection between the two.

Contextual Relevancy: 0.70 (Threshold: 0.5) - The score is 0.70 because the retrieval context has some relevant statements about software requirement characteristics, such as 'no ambiguo', 'conciso', 'consistente', 'completo', 'alcanzable', and 'verificable', but is largely irrelevant due to the presence of test cases and user stories that have no relation to the input.



## Caso de Prueba: 96 - Éxito

### Input del Usuario:

¿Cuáles son las características clave que deben tener los requerimientos de un producto de software para ser considerados válidos?

### Respuesta Generada:

Según el Contexto proporcionado, las características clave que deben tener los requerimientos de un producto de software para ser considerados válidos son: 1. **\*\*Necesario\*\***: Lo que se pide en un requisito debe ser necesario para el producto correcto. 2. **\*\*No ambiguo\*\***: El texto debe ser claro, preciso y tener una única interpretación posible. 3. **\*\*Conciso\*\***: Debe ser redactado en un lenguaje comprensible por los clientes, sin utilizar jerga técnica especializada. 4. **\*\*Consistente\*\***: Ningún requisito debe entrar en conflicto con otro requisito diferente, y el lenguaje empleado entre los distintos requisitos debe ser consistente. 5. **\*\*Completo\*\***: Los requisitos deben contener en sí mismos toda la información necesaria y no remitir a otras fuentes externas que los expliquen con más detalle. 6. **\*\*Alcanzable\*\***: Un requisito debe ser un objetivo realista, posible de ser alcanzado en el tiempo y los recursos disponibles. 7. **\*\*Verificable\*\***: Se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no, mediante inspección, análisis, demostración o testeo. Estas características clave se mencionan en el Documento 1 y se refieren a la calidad de los requisitos de software, que deben ser claros, precisos y no ambiguos para evitar problemas en el desarrollo del producto.

### Respuesta Esperada:

Según el contexto, las características clave que deben tener los requerimientos de un producto de software para ser considerados válidos son: \* No ambigua \* Comprobable o medible \* Necesaria para la aceptación del producto o proceso.

### Contexto Relevante:

1. REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitación es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the output mostly addresses the key characteristics of valid software product requirements, but slightly diverges by discussing their source.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node directly addresses the key characteristics of valid software product requirements, and the third-ranked node reiterates this, but the second-ranked node, which explains the process of specifying requirements, should be ranked lower as it does not explicitly state the key characteristics.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output can be directly attributed to nodes in the retrieval context, specifically nodes 1 and 3, which provide clear and concise definitions of software product requirements.

Contextual Relevancy: 0.84 (Threshold: 0.5) - The score is 0.84 because the retrieval context provides some relevant information about the characteristics of valid software product requirements, such as 'no ambiguous', 'concise', and 'verifiable', but is also cluttered with irrelevant information about the elicitation process, identification of actors and functionalities, and other software development stages.

## Caso de Prueba: 97 - Fallo

### Input del Usuario:

¿Cuál es el propósito de una restricción operativa en un requisito de software?

### Respuesta Generada:

Según el Documento 1, un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso. La restricción operativa en un requisito de software tiene como propósito definir las acciones o actividades que debe hacer o cumplir un sistema de software. En otras palabras, la restricción operativa en un requisito de software establece un límite o una condición que el sistema debe cumplir para satisfacer las necesidades del cliente. Por lo tanto, su propósito es especificar cómo debe comportarse el sistema en términos de funcionalidad, rendimiento, seguridad, entre otros aspectos, para garantizar que el sistema cumpla con los requisitos del cliente.

### Respuesta Esperada:

El propósito de una restricción operativa en un requisito de software es identificar una característica o limitación que es necesaria para la aceptación del producto o proceso.

### Contexto Relevante:

1. REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

3. arquitectura hexagonal puertos y adaptadores tipos de estilos arquitecturales para las aplicaciones web script archivo logica de la aplicacion la arquitectura hexagonal es un estilo arquitectural de p.....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the output partially addresses the purpose of an operational restriction, but gets sidetracked by discussing definitions and client requirements, which are not directly relevant to the question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 and node 2, are correctly ranked higher than irrelevant nodes, like node 3, which discusses 'arquitectura

hexagonal puertos y adaptadores' and is unrelated to operational restrictions in software requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly aligns with the 1st node in retrieval context, which explicitly mentions the necessary characteristic or operational restriction for product or process acceptance.

Contextual Relevancy: 0.35 (Threshold: 0.5) - The score is 0.35 because the retrieval context is mostly irrelevant to the input, with reasons such as 'The retrieval context contained the information 'una característica' which is not directly related to the purpose of an operational restriction in a software requirement.' and 'The statement is talking about the hexagonal architecture and its components, which is not related to the purpose of an operational restriction in a software requirement.' However, there are some relevant statements like 'una restriccion operativa funcional o de diseno de un producto o proceso' and 'definen las acciones o actividades que debe hacer o cumplir un sistema de software' that slightly increase the score.

## Caso de Prueba: 98 - Éxito

### Input del Usuario:

¿Cuáles son los tres aspectos clave que debe cumplir un requisito de software para ser considerado válido?

### Respuesta Generada:

Según el contexto proporcionado, específicamente en el Documento 4 y Documento 5, un requisito de software debe cumplir con tres aspectos clave para ser considerado válido: 1. **\*\*No ambiguo\*\***: El requisito debe ser claro y preciso, sin dejar espacio para interpretaciones erróneas. Debe tener una única interpretación posible. 2. **\*\*Comprobable o medible\*\***: El requisito debe ser verificable, es decir, debe ser posible determinar si el requisito se ha cumplido o no. 3. **\*\*Necesario para la aceptación del producto o proceso\*\***: El requisito debe ser esencial para que el software sea aceptable por el cliente, es decir, debe ser una característica o restricción operativa, funcional o de diseño necesaria para el éxito del proyecto. En resumen, un requisito de software debe ser claro, verificable y necesario para ser considerado válido.

### Respuesta Esperada:

Según el contexto, los tres aspectos clave que debe cumplir un requisito de software para ser considerado válido son: 1. No ambiguo 2. Comprobable o medible 3. Necesario para la aceptación del producto o proceso.

### Contexto Relevante:

1. REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitación es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación.....

### Métricas de Evaluación:

Answer Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because although the response addresses the input question, it contains redundant information that doesn't add new value, preventing it from being more relevant.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which explains the characteristics of software requirements, and node 2 which further supports the answer, are ranked higher than the irrelevant node 3 which discusses the process of specifying requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output can be directly attributed to the nodes in the retrieval context, specifically nodes 1 and 2, which provide clear and concise information about the characteristics of software requirements.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context provides relevant statements that directly answer the input question, such as 'necesario', 'no ambiguo', 'conciso', 'consistente', 'completo', 'alcanzable', and 'verificable', which are the key aspects a software requirement should fulfill to be considered valid, despite some irrelevant statements.

## Caso de Prueba: 99 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de los requisitos en cuanto a la aceptación del producto?

### Respuesta Generada:

Según el Documento 1, el propósito principal de los requisitos es definir lo que desea el cliente y especificar las acciones o actividades que debe hacer o cumplir un sistema de software, lo que es necesario para la aceptación del producto o proceso. En otras palabras, los requisitos identifican las características o restricciones operativas, funcionales o de diseño de un producto o proceso que son necesarias para la aceptación del producto.

### Respuesta Esperada:

El propósito principal de los requisitos en cuanto a la aceptación del producto es que sean necesarios para la aceptación del producto.

### Contexto Relevante:

1. REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restricción operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

2. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node, which clearly explains the main purpose of requirements, ranked highest.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, ensuring a flawless acceptance of the product.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context mostly talks about the definition and validation of requirements, but not specifically about their purpose in product acceptance, as stated in the input, and even contains irrelevant information like 'recorrido de bpm'.



## Caso de Prueba: 100 - Fallo

### Input del Usuario:

¿Cuál es el beneficio principal de definir adecuadamente los requerimientos en un proyecto en cuanto a la estimación de costos y tiempo de desarrollo?

### Respuesta Generada:

Según el Documento 4, el proceso de especificar requerimientos sirve para determinar los requerimientos de un sistema de software. En este proceso, la elicitation es fundamental para recopilar y descubrir los requerimientos del sistema a través de la comunicación directa con los interesados. A su vez, la especificación define qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. En este sentido, definir adecuadamente los requerimientos en un proyecto permite entender claramente el objetivo del proyecto y, en consecuencia, realizar una estimación más precisa de los costos y tiempo de desarrollo. De esta manera, se puede crear un plan de trabajo más efectivo y establecer prioridades adecuadas para cumplir con los objetivos del proyecto. En otras palabras, el beneficio principal de definir adecuadamente los requerimientos es que permite una mayor precisión en la estimación de costos y tiempo de desarrollo, lo que a su vez permite una mejor planificación y gestión del proyecto.

### Respuesta Esperada:

Definir adecuadamente los requerimientos permitirá realizar la estimación de costos y tiempo de desarrollo.

### Contexto Relevante:

1. Entender claramente cual es el objetivo de un proyecto Realizar la estimación de costos y tiempo de desarrollo  
Crear un plan de trabajo Establecer prioridades Definir adecuadamente los requerimientos  
.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos  
.....
2. iniciando un proyecto de software usando practicas agiles principales actividades en el sprint cero idea del sistema a desarrollar formar del equipo de desarrollo y definir roles fechas criticas neces.....
3. iniciando un proyecto de software usando practicas agiles principales actividades en el sprint cero formar del equipo de desarrollo y definir roles se presenta la idea del sistema a desarrollar fechas.....

### Métricas de Evaluación:

Answer Relevancy: 0.20 (Threshold: 0.5) - The score is 0.20 because the actual output contains multiple irrelevant statements that deviate from the input's focus on cost and time estimation, resulting in a low relevancy score.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which clearly addresses the question, are ranked higher than the irrelevant nodes, like the second and third nodes which are not directly related to the topic of defining requirements for cost and time estimation.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.19 (Threshold: 0.5) - The score is 0.19 because the retrieval context mostly talks about unrelated topics such as understanding project objectives, creating work plans, and setting priorities, as stated in the reasons for irrelevancy, whereas only a few statements like 'definir adecuadamente los requerimientos permitira' and 'realizar la estimacion de costos y tiempo de desarrollo' are relevant to the input.

## Caso de Prueba: 101 - Éxito

### Input del Usuario:

Definir adecuadamente los requerimientos permitirá reducir el tiempo de desarrollo de un proyecto de qué manera?

### Respuesta Generada:

Según el Documento 6, "el proceso para especificar requerimientos sirve para determinar los requerimientos" y "especificación define qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación". Al definir adecuadamente los requerimientos, se evitan malentendidos y ambigüedades en la comprensión de los requisitos del proyecto, lo que a su vez reduce la posibilidad de errores y rework durante el desarrollo del software. De acuerdo con el Documento 2, "requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios" y, por lo tanto, pueden generar problemas durante el desarrollo del proyecto. Al definir claramente los requerimientos, se evita esta ambigüedad y se pueden establecer prioridades claras para el desarrollo del proyecto. Además, según el Documento 1, "entender claramente cuál es el objetivo de un proyecto, realizar la estimación de costos y tiempo de desarrollo, crear un plan de trabajo establecer prioridades" permitirá una mayor eficiencia en el desarrollo del proyecto. Por lo tanto, definir adecuadamente los requerimientos permitirá reducir el tiempo de desarrollo de un proyecto de varias maneras: \* Evitando errores y rework debido a malentendidos o ambigüedades en los requisitos del proyecto. \* Estableciendo prioridades claras para el desarrollo del proyecto, lo que permite una planificación más eficiente del tiempo y los recursos. \* Reduciendo la posibilidad de cambiar los requisitos del proyecto durante el desarrollo, lo que puede generar demoras y aumentar los costos. En resumen, definir adecuadamente los requerimientos es crucial para reducir el tiempo de desarrollo de un proyecto de software, ya que permite una comprensión clara de los objetivos del proyecto, evita errores y ambigüedades, y establece prioridades claras para el desarrollo del proyecto.

### Respuesta Esperada:

Definir adecuadamente los requerimientos permitirá reducir el tiempo de desarrollo de un proyecto evitando errores y cambios costosos en etapas avanzadas del proyecto.

### Contexto Relevante:

1. Entender claramente cual es el objetivo de un proyecto Realizar la estimación de costos y tiempo de desarrollo  
Crear un plan de trabajo Establecer prioridades Definir adecuadamente los requerimientos  
.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo  
crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos  
.....

2. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision  
requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

3. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy día interactuar con los stake.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is highly relevant, but slightly digresses from the specific topic of defining requirements reducing development time, to discuss project management in general.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which mentions 'definir adecuadamente los requerimientos permitira', and node 2 which highlights the importance of defining requirements correctly, are ranked higher than the irrelevant node 3 which only has a broader topic of 'ciclo de vida del desarrollo de software', demonstrating perfect contextual precision.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire sentence in the expected output is perfectly matched with the 1st node in the retrieval context, resulting in a flawless recall.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context is mostly irrelevant, with reasons such as 'The statement is too broad and doesn't directly relate to defining project requirements' and 'The statement is about designing the software architecture, which is a different phase of the software development cycle', but some relevant statements like 'definir adecuadamente los requerimientos permitira' and 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision' managed to pull the score up.

## Caso de Prueba: 102 - Fallo

### Input del Usuario:

¿Cuál es el beneficio principal de definir adecuadamente los requerimientos en un proyecto?

### Respuesta Generada:

Según los documentos proporcionados, especialmente el Documento 2, el beneficio principal de definir adecuadamente los requerimientos en un proyecto es garantizar que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente. Al definir claramente los requerimientos, se evitan problemas de comunicación y se asegura que los desarrolladores entiendan correctamente lo que se necesita, lo que a su vez evita errores y retrasos durante el desarrollo del proyecto.

### Respuesta Esperada:

Definir adecuadamente los requerimientos permitirá Entender claramente cual es el objetivo de un proyecto.

### Contexto Relevante:

1. Entender claramente cual es el objetivo de un proyecto Realizar la estimación de costos y tiempo de desarrollo Crear un plan de trabajo Establecer prioridades Definir adecuadamente los requerimientos  
.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos  
.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
3. imprecision en los requerimientos asi fueron definidos los requerimientos asi lo entendieron los desarrolladores asi fue resuelto el problema anteriormente asi fue resuelto el problema ahora este es e.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2's lengthy description of the process and node 3's discussion of problems arising from undefined requirements, are correctly ranked lower than the highly relevant node 1 that directly answers the input question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in the retrieval context, capturing the essence of defining project requirements.

Contextual Relevancy: 0.37 (Threshold: 0.5) - The score is 0.37 because the retrieval context mostly contains irrelevant information, such as the elicitation process, identifying actors and functionalities, and problem descriptions, which do not directly address the main benefit of defining requirements in a project, as stated in the input.

## Caso de Prueba: 103 - Éxito

### Input del Usuario:

¿Cuál es la importancia de establecer prioridades en un proyecto?

### Respuesta Generada:

Según el Documento 2, la priorización es un principio fundamental en el desarrollo de software ágil, y se considera una parte vital del proyecto. Esto se debe a que la priorización permite evaluar un grupo de elementos y clasificarlos en orden de importancia o necesidad. Establecer prioridades en un proyecto es importante porque permite: \* Focalizar los esfuerzos en las tareas más importantes y necesarias para alcanzar el objetivo del proyecto. \* Asegurar que se estén trabajando en las historias de usuario más relevantes y urgentes. \* Optimizar el uso de los recursos y el tiempo de desarrollo. \* Reducir la complejidad y el riesgo del proyecto. \* Mejorar la comunicación y la colaboración entre el equipo de desarrollo, el product owner y los stakeholders. Además, como se menciona en el Documento 1, establecer prioridades es una de las etapas clave en la planificación de un proyecto, junto con understanding clearly the project objective, doing cost and time estimation, creating a work plan, and defining the requirements.

### Respuesta Esperada:

La importancia de establecer prioridades en un proyecto es que permite definir adecuadamente los requerimientos.

### Contexto Relevante:

1. Entender claramente cual es el objetivo de un proyecto Realizar la estimación de costos y tiempo de desarrollo Crear un plan de trabajo Establecer prioridades Definir adecuadamente los requerimientos  
.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos  
.....

2. priorizacion de las historias de usuario practicas agiles en scrum que es priorizar la priorizacion como principio significa hacer lo primero como proceso significa evaluar un grupo de elementos y cla.....

3. proyecto del curso objetivo del proyecto desarrollar un sistema de manejo de atencion a usuarios que permita coordinar y gestionar la atencion en multiples puntos de atencion adaptandose a actividades.....

### Métricas de Evaluación:

Answer Relevancy: 0.69 (Threshold: 0.5) - The score is 0.69 because while the output addresses project planning, it diverts to different aspects like project objective, cost estimation, work plan, and

requirements, instead of focusing solely on the importance of establishing priorities.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are ranked higher than the irrelevant node 3, which is about a specific project and does not provide any insight into the importance of establishing priorities in a project.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless alignment between the two.

Contextual Relevancy: 0.79 (Threshold: 0.5) - The score is 0.79 because the retrieval context has some relevant statements like 'establecer prioridades', 'la priorizacion de requerimientos es especialmente importante en el desarrollo de software agile', and 'priorizar la priorizacion como principio' that relate to the importance of establishing priorities in a project, but is brought down by irrelevant information like 'historia numbers' and 'proyecto del curso objetivo del proyecto'.



## Caso de Prueba: 104 - Fallo

### Input del Usuario:

¿Cuáles son las fuentes más comunes de errores en el desarrollo de software y cómo se pueden minimizar mediante un análisis de requisitos efectivo?

### Respuesta Generada:

Según el Documento 7, una de las fuentes más comunes de errores en el desarrollo de software son los requerimientos no precisos. Cuando los requerimientos no están declarados con precisión, pueden interpretarse de diferentes maneras por los desarrolladores y usuarios, lo que puede dar lugar a problemas en el desarrollo. Otras fuentes de errores en el desarrollo de software pueden incluir: \* Requerimientos ambiguos o vagos (Documento 7) \* Falta de comprensión de las necesidades del usuario (Documento 4) \* Falta de claridad en los requisitos (Documento 8) \* Inconsistencias en los requisitos (Documento 8) \* Requerimientos no alcanzables o no verificables (Documento 8) Para minimizar estos errores, es fundamental realizar un análisis de requisitos efectivo. Esto implica: \* Realizar una recolección exhaustiva de requisitos (Documento 5) \* Establecer un lenguaje común entre los stakeholders (Documento 8) \* Redactar requisitos claros, precisos y no ambiguos (Documento 8) \* Verificar la consistencia y completitud de los requisitos (Documento 8) \* Priorizar y gestionar los requisitos de manera efectiva (Documento 6) \* Realizar una validación de los requisitos con los usuarios y expertos (Documento 10) \* Utilizar técnicas de prototipado y recorrido de BPM para validar los requisitos (Documento 10) Al realizar un análisis de requisitos efectivo, se pueden minimizar los errores en el desarrollo de software y garantizar que el producto final cumpla con las necesidades del usuario.

### Respuesta Esperada:

Según el contexto, las fuentes más comunes de errores en el desarrollo de software provienen de la falta de claridad en los requisitos del proyecto. Esto puede deberse a una mala comunicación entre los stakeholders, la ambigüedad en la definición de los requisitos, la falta de documentación o la no consideración de los requisitos no funcionales. Para minimizar estos errores, es fundamental realizar un análisis de requisitos efectivo, que implica: \* Establecer una comunicación clara y abierta con los stakeholders para entender sus necesidades y expectativas. \* Definir requisitos claros, precisos y no ambiguos. \* Documentar exhaustivamente los requisitos y asegurarse de que todos los involucrados los entiendan. \* Considerar todos los requisitos, incluyendo los no funcionales, como rendimiento, seguridad y escalabilidad. \* Realizar una revisión y validación exhaustiva de los requisitos para detectar y corregir errores tempranamente. De esta manera, se puede reducir significativamente el número de errores en el desarrollo de software y asegurar que el producto final cumpla con los requisitos y expectativas de los usuarios.

### Contexto Relevante:

1. ¿De donde provienen los errores en el Desarrollo de software?...

### Contexto Recuperado:

1. de donde provienen los errores en el desarrollo de software...

2. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....

3. 1xx mensajes informativos no se utilizan y estan reservados para un futuro uso 3xx mensajes de redireccion que informan de operaciones complementarias que se deben realizar para finalizar la operacion.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a comprehensive and relevant response without any irrelevant statements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 with 'estimacion de tiempos y costos en marco de las metodologias agiles' and node 3 with 'codigos de error en paginas web', are correctly ranked lower than the relevant node 1, which directly addresses the input question.

Contextual Recall: 0.43 (Threshold: 0.5) - The score is 0.43 because the model struggled to attribute sentences 3-5 in the expected output to the nodes in retrieval context, but successfully linked sentences 1-2 to the 3rd node, which discusses 'errores en el desarrollo de software'.

Contextual Relevancy: 0.07 (Threshold: 0.5) - The score is 0.07 because the retrieval context is mostly irrelevant, with reasons stating it talks about estimation, costs, and HTTP status codes, which are not directly related to common sources of errors in software development, and only a small part of it mentions 'de donde provienen los errores en el desarrollo de software' which is somewhat relevant.

## Caso de Prueba: 105 - Fallo

### Input del Usuario:

¿Cuáles son las fuentes más comunes de errores durante el desarrollo de software?

### Respuesta Generada:

Según el Documento 1, las fuentes más comunes de errores durante el desarrollo de software no se mencionan explícitamente. Sin embargo, se puede inferir que los errores pueden surgir en diferentes etapas del ciclo de vida del desarrollo de software, como se muestra en el Documento 2. Algunas de las fuentes de errores posibles podrían ser: \* Imprecisión en los requerimientos (Documento 10): los requerimientos ambiguos o no claros pueden llevar a errores de interpretación por parte de los desarrolladores. \* Fallos en la estimación de tiempos y costos (Documento 3): una estimación inexacta de los recursos necesarios puede llevar a delays y errores en el proyecto. \* Errores en la creación de la arquitectura del sistema (Documento 2): una mala diseño o arquitectura del sistema puede llevar a problemas de rendimiento, escalabilidad y mantenimiento. \* Errores en la fase de desarrollo (Documento 2): errores en la codificación, pruebas y despliegue del sistema pueden llevar a bugs y problemas de funcionamiento. \* Fallos en la evaluación de la calidad del sistema (Documento 2): no detectar errores o problemas de calidad durante la fase de pruebas y evaluación puede llevar a errores en la producción. Es importante destacar que estos son solo algunos posibles ejemplos de fuentes de errores durante el desarrollo de software, y que la lista no es exhaustiva.

### Respuesta Esperada:

Las fuentes más comunes de errores durante el desarrollo de software provienen del propio proceso de desarrollo.

### Contexto Relevante:

1. ¿De donde provienen los errores en el Desarrollo de software?...

### Contexto Recuperado:

1. de donde provienen los errores en el desarrollo de software...
2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy día interactuar con los stake.....
3. estimacion de tiempos y costos en marco de las metodologias agiles estimacion es una aproximacion informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....

### Métricas de Evaluación:

Answer Relevancy: 0.89 (Threshold: 0.5) - The score is 0.89 because although the response addresses the question, it includes a negative statement that doesn't provide any common sources of errors, which prevents it from being a perfect match.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which directly matches the input question and node 2 which explains the development process, are ranked higher than the irrelevant node 3, which talks about estimation and costs.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in the retrieval context, which directly mentions software development errors, resulting in a flawless recall.

Contextual Relevancy: 0.11 (Threshold: 0.5) - The score is 0.11 because the retrieval context is mostly irrelevant, with reasons such as 'The statement is too broad and doesn't specifically address common sources of errors during software development' and 'The statement is about estimation of time and cost in agile methodologies, which is not directly related to common sources of errors during software development', but has some relevant statements like 'de donde provienen los errores en el desarrollo de software' and 'evaluar la calidad del sistema para encontrar y corregir posibles errores'.

## Caso de Prueba: 106 - Fallo

### Input del Usuario:

¿Cuáles son las fuentes más comunes de errores durante el desarrollo de software?

### Respuesta Generada:

Según el Documento 1, las fuentes más comunes de errores durante el desarrollo de software no se especifican explícitamente. Sin embargo, se puede inferir que los errores pueden surgir en diferentes etapas del ciclo de vida del desarrollo de software, como se describe en el Documento 2. En particular, algunos errores comunes pueden surgir en las siguientes etapas: 1. Requerimientos: Imprecisiones en los requerimientos (Documento 10) pueden llevar a errores en el desarrollo. 2. Diseño: Un diseño deficiente o incorrecto puede generar errores en el sistema. 3. Desarrollo: Errores en la codificación, pruebas y depuración pueden generar problemas en el sistema. 4. Pruebas: Fallos en la evaluación de la calidad del sistema pueden dejar pasar errores. 5. Despliegue: Problemas en la implementación y configuración del sistema pueden generar errores. Es importante destacar que la estimación de tiempos y costos (Documento 3) también puede influir en la calidad del desarrollo y la presencia de errores. Si se estima incorrectamente el tiempo y los recursos necesarios, puede llevar a rushes en el desarrollo, lo que aumenta la probabilidad de errores. En resumen, aunque no se especifican explícitamente las fuentes más comunes de errores, se pueden identificar varios puntos críticos en el ciclo de vida del desarrollo de software donde pueden surgir errores.

### Respuesta Esperada:

Las fuentes más comunes de errores durante el desarrollo de software son: \* La propia comprensión del requisito o especificación del software \* La interpretación incorrecta de los requisitos del cliente \* La falta de experiencia o habilidades del desarrollador \* La complejidad del código \* La falta de pruebas adecuadas \* La mala comunicación entre los miembros del equipo de desarrollo \* La presión por cumplir con fechas límite \* La falta de documentación adecuada \* La mala gestión de cambios en el código \* La falta de revisión y depuración del código.

### Contexto Relevante:

1. ¿De donde provienen los errores en el Desarrollo de software?...

### Contexto Recuperado:

1. de donde provienen los errores en el desarrollo de software...
2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideación concepto ciclo de vida del software hoy día interactuar con los stakeholders.....
3. estimación de tiempos y costos en marco de las metodologías ágiles estimación es una aproximación informada basada en datos disponibles experiencia pasada y supuestos razonables costos directos son aq.....

### Métricas de Evaluación:

Answer Relevancy: 0.82 (Threshold: 0.5) - The score is 0.82 because although the output addresses the input question, it contains some irrelevant statements that are negations or repetitions of the input, preventing a perfect score.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, such as node 1 with the reason 'de donde provienen los errores en el desarrollo de software' is a vague question and doesn't provide any specific information about the common sources of errors, node 2 with the reason 'ciclo de vida del desarrollo de software...' is a passage about the software development life cycle and doesn't mention the common sources of errors, and node 3 with the reason 'estimacion de tiempos y costos...' is a passage about estimation of time and costs in agile methodologies and doesn't provide any information about the common sources of errors, are irrelevant to the input.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output can be attributed to any node in the retrieval context.

Contextual Relevancy: 0.12 (Threshold: 0.5) - The score is 0.12 because the retrieval context mostly talks about the software development life cycle, user requirements, and other unrelated topics, with only a few statements mentioning error sources, such as 'de donde provienen los errores en el desarrollo de software' and 'evaluar la calidad del sistema para encontrar y corregir posibles errores'.

## Caso de Prueba: 107 - Fallo

### Input del Usuario:

¿Cuál es la principal fuente de errores de codificación en el software?

### Respuesta Generada:

Según el contexto proporcionado, no se encuentra una respuesta directa a la pregunta sobre la principal fuente de errores de codificación en el software. Sin embargo, podemos inferir que los errores en el desarrollo de software pueden surgir de diversas fuentes, incluyendo: 1. Errores del cliente (4xx): El requerimiento contiene algún error o no puede ser realizado. 2. Errores del servidor (5xx): El servidor no ha podido llevar a cabo una solicitud. Además, los conceptos fundamentales de la ingeniería de software, como la programación y la metodología ágil de desarrollo de software, pueden influir en la calidad del código y la posibilidad de errores. En resumen, no hay una respuesta única y clara sobre la principal fuente de errores de codificación en el software, pero se pueden identificar varias fuentes potenciales de errores en el desarrollo de software.

### Respuesta Esperada:

La principal fuente de errores de codificación en el software es el propio desarrollo de software.

### Contexto Relevante:

1. ¿De donde provienen los errores en el Desarrollo de software?...

### Contexto Recuperado:

1. de donde provienen los errores en el desarrollo de software...
2. 1xx mensajes informativos no se utilizan y estan reservados para un futuro uso 3xx mensajes de redireccion que informan de operaciones complementarias que se deben realizar para finalizar la operacion.....
3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because although the output doesn't directly answer the question, it still provides some relevant context, but is brought down by the irrelevant statements that don't provide any information about the principal source of errors.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which is directly related to the question, is correctly ranked higher than the second node, which is about error codes in web pages and should be ranked lower. However, the score is not higher because the second node, which is irrelevant, is not ranked lower than the third node, which is also directly related to the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions software development errors, resulting in a flawless match.

Contextual Relevancy: 0.17 (Threshold: 0.5) - The score is 0.17 because the retrieval context is mostly irrelevant, with statements about HTTP status codes and software development methodologies, as stated in the reasons for irrelevancy, and only a few relevant statements like 'de donde provienen los errores en el desarrollo de software' and 'codigos de error desarrollo de software'.



## Caso de Prueba: 108 - Fallo

### Input del Usuario:

¿Cuáles son las principales fuentes de errores en el desarrollo de software?

### Respuesta Generada:

Según el Documento 2, no se proporciona una lista explícita de las principales fuentes de errores en el desarrollo de software. Sin embargo, podemos inferir que algunos de los problemas comunes en el desarrollo de software son: \* Problemas de calidad (Documento 7) \* Entregas tardías \* Sobrecosto fuera del presupuesto estimado Además, al analizar otros documentos, podemos identificar algunas fuentes de errores adicionales, tales como: \* Errores del cliente o del servidor (Documento 6) \* Problemas con la gestión de variables de sesión (Documento 8) \* Fallos en la validación de requisitos (Documento 9) \* Ineficiencias en la planificación y colaboración en equipo (Documento 10) Es importante destacar que la lista de fuentes de errores puede ser más amplia y variada, y puede depender del contexto específico del proyecto de desarrollo de software.

### Respuesta Esperada:

Las principales fuentes de errores en el desarrollo de software provienen de la propia naturaleza del proceso de desarrollo.

### Contexto Relevante:

1. ¿De donde provienen los errores en el Desarrollo de software?...

### Contexto Recuperado:

1. navegador web servidor web script script script scripts que corren del lado del servidor navegador web navegador web clientes y servidores web navegador web tarea cuales son las principales diferencia.....
2. de donde provienen los errores en el desarrollo de software...
3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarro.....

### Métricas de Evaluación:

Answer Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the output provides some relevant information, but gets sidetracked with general comments on software development that don't specifically address the main question about sources of error.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because only half of the top-ranked nodes in the retrieval contexts are relevant to the input, with the second node directly addressing the question, while the first and third nodes, ranked 1 and 3 respectively, are about unrelated topics such as 'navegador web servidor web...' and 'objetivos del curso...', which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, specifically the 2nd node, which shares similar keywords.

Contextual Relevancy: 0.22 (Threshold: 0.5) - The score is 0.22 because the retrieval context is mostly irrelevant, with statements discussing web browsers, servers, and scripts, and only a few mentions of 'desarrollo de software' and 'de donde provienen los errores en el desarrollo de software' that are actually related to the input.

## Caso de Prueba: 109 - Éxito

### Input del Usuario:

¿Cuál es la definición de un requerimiento en el contexto de un sistema o componente del sistema?

### Respuesta Generada:

Según el Documento 4, un requerimiento es "una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo... que define lo que el sistema debe hacer en cuanto a sus procesos, consultas, reportes". En otras palabras, un requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso que es no ambigua, comprobable, medible y necesaria para la aceptación del producto o proceso (Documento 2 y 3). En el contexto de un sistema o componente del sistema, un requerimiento especifica lo que desea el cliente y define las acciones o actividades que debe hacer o cumplir un sistema de software.

### Respuesta Esperada:

Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo, que satisface condiciones y define lo que se necesita en un sistema o componente del sistema.

### Contexto Relevante:

1. 3 COSAS QUE CUMPLEN LOS REQUERIMIENTOS Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Resuelve una necesidad Capacidad que debe e.....

### Contexto Recuperado:

1. ejercicio elaborar la descripción de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad el sistema debe permitir crear un usuario del sistema...
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because the output partially addresses the input question, but is brought down by repetitive and incomplete definitions that only provide characteristics of a requirement without fully defining it.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the node at rank 1 is irrelevant due to being about 'ejercicio elaborar la descripción de un requerimiento funcional...' which is not

directly related to defining what a requirement is in the context of a system or component, and the node at rank 3 is also irrelevant as it's about 'resumen' which does not provide any relevant information about the definition of a requirement in the context of a system or component, whereas the node at rank 2 is relevant as it provides a clear definition of a requirement.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which provides a clear definition of a requirement.

Contextual Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because the retrieval context is only partially relevant, as it provides some definitions of a requirement in the context of a system or component (e.g., 'un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso'), but also includes irrelevant information, such as exercises and categories.

## Caso de Prueba: 110 - Fallo

### Input del Usuario:

¿Cuál es la definición de un requerimiento en un sistema o componente del sistema?

### Respuesta Generada:

Según el Documento 4, un requerimiento es "una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo" que "resuelve una necesidad capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, un estándar, una especificación u otro documento formal". En otras palabras, un requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso que es no ambigua, comprobable o medible y necesaria para la aceptación del producto o proceso.

### Respuesta Esperada:

Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo, o una capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, un estándar, una especificación u otro documento formal.

### Contexto Relevante:

1. 3 COSAS QUE CUMPLEN LOS REQUERIMIENTOS Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Resuelve una necesidad Capacidad que debe e.....

### Contexto Recuperado:

1. ejercicio elaborar la descripción de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad el sistema debe permitir crear un usuario del sistema...
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.43 (Threshold: 0.5) - The score is 0.43 because the output only provides characteristics of a requirement, not its definition, which is the main focus of the input question.

Contextual Precision: 0.58 (Threshold: 0.5) - The score is 0.58 because the first node, which is not directly related to the definition of a requirement, is ranked higher than the second and third nodes, which provide clear definitions of a requirement.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which provides a clear definition of a requirement.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context only partially addresses the input question, with relevant statements like 'un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso' and 'los requerimientos especifican lo que desea el cliente y definen las acciones o actividades que debe hacer o cumplir un sistema de software', but is hindered by irrelevant phrases like 'requerimientos del software' and 'resumen'.

## Caso de Prueba: 111 - Éxito

### Input del Usuario:

¿Qué es un requisito en el contexto de un contrato?

### Respuesta Generada:

Según el Documento 2, en el contexto de un contrato, un requisito se define como "una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo". En otras palabras, un requisito es una capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, estandar, especificación u otro documento formal. Además, define lo que el sistema debe hacer en cuanto a sus procesos, consultas, reportes, y define lo que se necesita, que es un requerimiento.

### Respuesta Esperada:

En el contexto de un contrato, un requisito es una condición o necesidad que debe estar presente en un sistema o componentes del sistema para satisfacer el contrato.

### Contexto Relevante:

1. 3 COSAS QUE CUMPLEN LOS REQUERIMIENTOS Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Resuelve una necesidad Capacidad que debe e.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. cosas que cumplen los requerimientos una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo resuelve una necesidad capacidad que debe estar.....
3. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....

### Métricas de Evaluación:

Answer Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the response partially addresses the input question, but gets sidetracked by describing what a requirement does in a system, rather than focusing on its meaning in the context of a contract.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the top-ranked nodes providing direct definitions and explanations of a requirement in the context of a contract.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which mirrors the same concept of a

requirement being a condition or need that must be present in a system to satisfy a contract.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the input, providing a comprehensive explanation of what a requirement is in the context of a contract, covering aspects such as necessity, clarity, consistency, completeness, achievability, and verifiability.



## Caso de Prueba: 112 - Éxito

### Input del Usuario:

¿Qué es un requerimiento en el contexto de especificación de un sistema o componente?

### Respuesta Generada:

Según los documentos proporcionados, un requerimiento en el contexto de especificación de un sistema o componente se define como una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso que es no ambigua, comprobable o medible y necesaria para la aceptación del producto o proceso (Documento 2 y Documento 3). En otras palabras, un requerimiento especifica lo que desea el cliente y define las acciones o actividades que debe hacer o cumplir un sistema de software. Es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo, y resuelve una necesidad o capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, un estándar, una especificación u otro documento formal (Documento 4). En el contexto de especificación de un sistema o componente, los requerimientos pueden ser funcionales o no funcionales. Los requerimientos funcionales se refieren a las declaraciones de los servicios que prestara el sistema, como por ejemplo, "El sistema debe permitir crear un usuario del sistema" (Documento 6 y Documento 8). Los requerimientos no funcionales, por otro lado, se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como por ejemplo, limitaciones de tiempo, rendimiento, calidad, normas tecnológicas, etc. (Documento 10).

### Respuesta Esperada:

Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo, o una capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, un estándar, una especificación u otro documento formal.

### Contexto Relevante:

1. 3 COSAS QUE CUMPLEN LOS REQUERIMIENTOS Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Resuelve una necesidad Capacidad que debe e.....

### Contexto Recuperado:

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the response perfectly addresses the input question, providing a precise and relevant answer without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node explaining the context of writing requirements, the second node directly defining what a requirement is, and the third node being a duplicate of the second but still providing relevant information, resulting in a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which provides a matching definition of a requerimiento.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context has some relevant statements that define a requirement in the context of system or component specification, such as 'un requisito o requerimiento es una declaracion que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso', but is brought down by irrelevant information about design, software development life cycle, and summaries that do not provide direct information about the input.

## Caso de Prueba: 113 - Éxito

### Input del Usuario:

¿Qué es un requerimiento en el contexto de un proceso?

### Respuesta Generada:

Según los documentos proporcionados, un requerimiento en el contexto de un proceso es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso que es no ambigua, comprobable o medible y necesaria para la aceptación del producto o proceso. Los requerimientos especifican lo que desea el cliente y definen las acciones o actividades que debe hacer o cumplir un sistema de software. En otras palabras, un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo, que define lo que el sistema debe hacer en cuanto a sus procesos, consultas, reportes, y que satisface condiciones definidas en un contrato, estándar, especificación u otro documento formal. Por lo tanto, un requerimiento es una descripción detallada y precisa de lo que se necesita que haga un sistema de software para satisfacer las necesidades de los usuarios.

### Respuesta Esperada:

Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

### Contexto Relevante:

1. 3 COSAS QUE CUMPLEN LOS REQUERIMIENTOS Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Resuelve una necesidad Capacidad que debe e.....

### Contexto Recuperado:

1. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

3. ejercicio elaborar la descripcion de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad el sistema debe permitir crear un usuario del sistema...

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the response perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which provide clear definitions of a requirement, are ranked higher than the irrelevant node 3, which is an exercise prompt and does not directly address the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the definition of a requirement provided in the 1st node in the retrieval context.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context contains some relevant statements that define what a requirement is in the context of a process, such as 'un requisito o requerimiento es una declaracion que identifica una caracteristica o restricción operativa funcional o de diseño de un producto o proceso', but is brought down by irrelevant phrases like 'requerimientos del software' and 'resumen' that do not provide a clear definition.

## Caso de Prueba: 114 - Fallo

### Input del Usuario:

¿Cuál es la diferencia entre un requisito funcional y un requisito no funcional en el análisis de requisitos de un sistema?

### Respuesta Generada:

Según los documentos proporcionados, la diferencia entre un requisito funcional y un requisito no funcional en el análisis de requisitos de un sistema se puede definir de la siguiente manera: **\*\*Requisito Funcional\*\***: Un requisito funcional es una declaración que describe el comportamiento del sistema en condiciones específicas. Es decir, establece lo que el sistema debe hacer en respuesta a determinados insumos. Los requisitos funcionales describen los servicios que prestará el sistema y cómo reaccionará a ciertos eventos o condiciones. Por ejemplo, "El sistema debe permitir crear un usuario del sistema" es un requisito funcional. **\*\*Requisito No Funcional\*\***: Un requisito no funcional, por otro lado, establece restricciones o límites que el sistema debe cumplir, pero no se refiere a la funcionalidad específica del sistema. Estos requisitos se centran en la performance, seguridad, usabilidad, compatibilidad, entre otros aspectos, que no están directamente relacionados con la funcionalidad del sistema. Por ejemplo, "El sistema debe responder a una solicitud en un máximo de 2 segundos" es un requisito no funcional. En resumen, los requisitos funcionales se centran en lo que el sistema debe hacer, mientras que los requisitos no funcionales se centran en cómo debe hacerlo y qué restricciones debe cumplir. Es importante destacar que tanto los requisitos funcionales como los no funcionales son importantes para la correcta definición de los requisitos del sistema y para asegurar que se cumplan las necesidades del cliente.

### Respuesta Esperada:

En el análisis de requisitos de un sistema, la diferencia entre un requisito funcional y un requisito no funcional es la siguiente: \* Un requisito funcional se refiere a lo que el sistema debe hacer, es decir, las funciones o tareas que debe realizar. Estos requisitos describen el comportamiento del sistema y se centran en lo que el sistema debe hacer para satisfacer las necesidades del usuario. \* Un requisito no funcional, por otro lado, se refiere a cómo el sistema debe realizar sus funciones, es decir, las restricciones y limitaciones que se imponen al sistema. Estos requisitos describen las características del sistema que no están directamente relacionadas con su función, pero que son importantes para su éxito, como la seguridad, el rendimiento, la escalabilidad, la usabilidad, entre otros.

### Contexto Relevante:

1. Clasificación de los requerimientos....

### Contexto Recuperado:

1. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....

2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....

3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts clearly explains the difference between functional and non-functional requirements, and the third-ranked node provides a relevant definition, but the second-ranked node, which discusses HTTP requests, is an irrelevant node that should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, with the 1st node accurately explaining the difference between functional and non-functional requirements.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context is mostly irrelevant, with statements repeatedly mentioning HTTP methods, which 'is not related to the input about functional and non-functional requirements', but a few relevant statements were found, such as 'requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema', which shows some connection to the input.

## Caso de Prueba: 115 - Éxito

### Input del Usuario:

¿Cuál es la diferencia entre requerimientos funcionales y no funcionales en la clasificación de requisitos?

### Respuesta Generada:

Según los documentos proporcionados, la diferencia entre requerimientos funcionales y no funcionales es la siguiente: **\*\*Requerimientos Funcionales\*\***: Son aquellos que se refieren directamente a las funciones específicas que debe entregar el sistema. Describen el comportamiento del sistema en condiciones específicas, es decir, cómo reacciona a determinados insumos. Estos requieren especificar lo que el sistema debe hacer. (Documento 2, 5, 8 y 10) **\*\*Requerimientos No Funcionales\*\***: No se refieren directamente a las funciones específicas del sistema, sino a las propiedades emergentes del mismo, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. Establecen restricciones en el sistema, como la capacidad de los dispositivos de entrada/salida y la representación de datos en la interfaz del sistema. También se conocen como requisitos técnicos, atributos de calidad o requisitos de calidad de servicio. (Documento 1, 3, 5, 6 y 7) En resumen, los requerimientos funcionales se enfocan en lo que el sistema debe hacer, mientras que los requerimientos no funcionales se enfocan en cómo debe hacerlo, estableciendo restricciones y límites en el sistema.

### Respuesta Esperada:

En la clasificación de requisitos, los requerimientos funcionales se refieren a aquellos que describen lo que el sistema debe hacer, es decir, las funciones que debe realizar. Por otro lado, los requerimientos no funcionales se refieren a cómo el sistema debe realizar esas funciones, es decir, las restricciones o características que deben cumplir.

### Contexto Relevante:

1. Clasificación de los requerimientos....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....

3. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise distinction between functional and non-functional requirements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the top 3 nodes directly addressing the expected output, providing clear explanations for functional and non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, with the 1st node matching 'las funciones que debe realizar' and the 2nd and 3rd nodes matching 'las restricciones o características que deben cumplir'.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context provides a clear and comprehensive explanation of the difference between functional and non-functional requirements, perfectly aligning with the input question.



## Caso de Prueba: 116 - Éxito

### Input del Usuario:

¿Cuál es la principal característica que define a un requisito funcional en la clasificación de los requerimientos?

### Respuesta Generada:

Según el Documento 7, la principal característica que define a un requisito funcional es que declara "los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la forma en que el sistema debe comportarse en situaciones particulares". En otras palabras, un requisito funcional se centra en describir el comportamiento del sistema en condiciones específicas, es decir, qué debe hacer el sistema y cómo debe reaccionar a ciertas entradas o situaciones.

### Respuesta Esperada:

La principal característica que define a un requisito funcional es que describe lo que el sistema debe hacer.

### Contexto Relevante:

1. Clasificación de los requerimientos....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and accurate response without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node mentioning 'describe lo que el sistema debe hacer', the second node directly relating to the expected output, and the third node repeating the relevant information, resulting in a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in the retrieval context, which clearly defines a functional requirement.

Contextual Relevancy: 0.73 (Threshold: 0.5) - The score is 0.73 because the retrieval context provides relevant statements that define a functional requirement, such as 'un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso', but is brought down by irrelevant phrases like 'los requerimientos del software' that are too broad.

## **Caso de Prueba: 117 - Éxito**

### **Input del Usuario:**

¿Cuál es la principal diferencia entre los requisitos funcionales y no funcionales en la clasificación de los requerimientos de un sistema?

### **Respuesta Generada:**

Según los documentos proporcionados, la principal diferencia entre los requisitos funcionales y no funcionales es que los requisitos funcionales se refieren a las funciones específicas que debe proporcionar el sistema, como la forma en que reacciona a las entradas y se comporta en situaciones particulares, mientras que los requisitos no funcionales se refieren a las propiedades emergentes del sistema, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento, que imponen restricciones en las funciones del sistema. En otras palabras, los requisitos funcionales describen lo que el sistema debe hacer, mientras que los requisitos no funcionales describen cómo debe hacerlo. Los requisitos no funcionales también se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como limitaciones en el proceso de desarrollo, rendimiento, calidad, normas tecnológicas, etc. Por ejemplo, un requisito funcional podría ser "el sistema debe permitir crear un usuario del sistema", mientras que un requisito no funcional podría ser "el sistema debe responder a la solicitud de creación de usuario en un plazo de 2 segundos".

### **Respuesta Esperada:**

La principal diferencia entre los requisitos funcionales y no funcionales es que los primeros se refieren a lo que el sistema debe hacer, es decir, las funciones que debe realizar, mientras que los segundos se refieren a cómo debe hacerlo, es decir, las restricciones y características que debe cumplir.

### **Contexto Relevante:**

1. Clasificación de los requerimientos....

### **Contexto Recuperado:**

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....
3. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and relevant answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all retrieval context nodes, ranked 1 to 3, are highly relevant to the input, providing clear explanations and definitions of functional and non-functional requirements, making it a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with nodes 1, 2, and 3 in the retrieval context, which explicitly mention key concepts like 'las funciones que debe realizar' and 'restricciones y características que debe cumplir'.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context has some irrelevant statements about 'interface del sistema' and 'reglamentos de seguridad', but it also provides clear and direct explanations of functional and non-functional requirements, such as 'los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema' and 'requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema'.

## Caso de Prueba: 118 - Fallo

### Input del Usuario:

¿Cuál es la diferencia entre un requisito funcional y un requisito no funcional en el modelado de requisitos?

### Respuesta Generada:

Según los documentos proporcionados, la diferencia entre un requisito funcional y un requisito no funcional en el modelado de requisitos es la siguiente: **\*\*Requisito Funcional\*\***: Un requisito funcional es una declaración que describe el servicio que prestará el sistema, es decir, cómo reaccionará a determinados insumos. Describe el comportamiento del sistema en condiciones específicas. Por ejemplo, en el Documento 5, se menciona un requisito funcional para el proceso de registro de un curso en el sistema de la universidad, donde el sistema debe permitir crear un usuario del sistema. **\*\*Requisito No Funcional\*\***: Un requisito no funcional, por otro lado, especifica cómo debe comportarse un sistema, imponiendo restricciones que limitan los requisitos funcionales. También conocidos como requisitos técnicos, atributos de calidad o requisitos de calidad de servicio. No describen lo que el sistema debe hacer, sino cómo debe hacerlo. En resumen, los requisitos funcionales se centran en lo que el sistema debe hacer, mientras que los requisitos no funcionales se centran en cómo debe hacerlo.

### Respuesta Esperada:

En el modelado de requisitos, la diferencia entre un requisito funcional y un requisito no funcional radica en su propósito y enfoque. Un requisito funcional se refiere a una característica o capacidad que el sistema debe proporcionar para cumplir con los objetivos del negocio o usuario. Estos requisitos describen lo que el sistema debe hacer, es decir, las funciones que debe realizar. Por ejemplo, "el sistema debe permitir a los usuarios registrarse con una contraseña" es un requisito funcional. Por otro lado, un requisito no funcional se refiere a una restricción o característica que no se relaciona directamente con la funcionalidad del sistema, pero que afecta su desempeño, seguridad, escalabilidad, entre otros aspectos. Estos requisitos describen cómo el sistema debe funcionar, es decir, las características no relacionadas con la funcionalidad. Por ejemplo, "el sistema debe responder a las solicitudes en menos de 2 segundos" es un requisito no funcional. En resumen, los requisitos funcionales se centran en lo que el sistema debe hacer, mientras que los requisitos no funcionales se centran en cómo lo hace.

### Contexto Relevante:

1. Clasificación de los requerimientos....

### Contexto Recuperado:

1. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....

2. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....

3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise distinction between functional and non-functional requirements in requirement modeling!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1, which explains the difference between functional and non-functional requirements, are ranked higher than the irrelevant nodes, like node 2 and node 3, which do not specifically address the difference between functional and non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to nodes in the retrieval context, such as sentence 1 being supported by the 2nd node, sentence 2 by the 3rd node, and sentences 3-4 by the 1st node, demonstrating a flawless contextual recall.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context mostly lacks relevant information about functional and non-functional requirements, as stated in the reasons for irrelevancy, and only provides a few relevant statements, such as 'requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema', which is not enough to make the context highly relevant to the input.

## Caso de Prueba: 119 - Éxito

### Input del Usuario:

¿Cuál es la principal diferencia entre los requerimientos funcionales y no funcionales en la clasificación de los requerimientos de un sistema?

### Respuesta Generada:

La principal diferencia entre los requerimientos funcionales y no funcionales es que los requerimientos funcionales describen las funciones específicas que debe realizar un sistema, es decir, se refieren a lo que el sistema debe hacer, mientras que los requerimientos no funcionales se centran en las propiedades emergentes del sistema, como la fiabilidad, el rendimiento, la seguridad, entre otras, que limitan o restringen la forma en que el sistema debe funcionar. En otras palabras, los requerimientos funcionales se enfocan en lo que el sistema debe hacer, mientras que los requerimientos no funcionales se enfocan en cómo lo debe hacer. Por ejemplo, un requerimiento funcional podría ser "el sistema debe permitir al usuario iniciar sesión con su nombre de usuario y contraseña", mientras que un requerimiento no funcional podría ser "el sistema debe responder a las solicitudes de inicio de sesión en menos de 2 segundos". En resumen, los requerimientos funcionales se centran en la funcionalidad del sistema, mientras que los requerimientos no funcionales se centran en las características de calidad y restricciones del sistema.

### Respuesta Esperada:

La principal diferencia entre los requerimientos funcionales y no funcionales es que los primeros se refieren a las declaraciones de los servicios que debe proporcionar el sistema y cómo debe comportarse en situaciones particulares, mientras que los segundos se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como tiempo, rendimiento, calidad, normas, tecnología, etc.

### Contexto Relevante:

1. Clasificación de los requerimientos. Los requerimientos funcionales •Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la for.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos no funcionales técnicas para validación de requerimientos 1 validación de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....
3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 is highly relevant, and node 3 is also relevant, but node 2, which is about validation techniques, is an irrelevant node that should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes 1 and 3 in the retrieval context, which clearly describe functional and non-functional requirements, resulting in a flawless match.

Contextual Relevancy: 0.54 (Threshold: 0.5) - The score is 0.54 because the retrieval context mostly lacks information about the difference between functional and non-functional requirements, as stated multiple times, but has some relevant statements that provide definitions and characteristics of both types of requirements.



## Caso de Prueba: 120 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre los requerimientos funcionales y no funcionales en la clasificación de los requerimientos de un proyecto?

### Respuesta Generada:

Según los documentos proporcionados, la principal diferencia entre los requerimientos funcionales y no funcionales es su enfoque. Los requerimientos funcionales se refieren a las funciones específicas que debe proporcionar el sistema, es decir, describen el comportamiento del sistema en condiciones específicas. Estos requerimientos definen los servicios que debe prestar el sistema y cómo debe reaccionar a las entradas. Por otro lado, los requerimientos no funcionales no se refieren directamente a las funciones específicas del sistema, sino a las propiedades emergentes del mismo, como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, entre otras. Estos requerimientos imponen restricciones en el sistema, limitando las funciones o servicios que se ofrecen, y se relacionan con la calidad del sistema, su rendimiento, la tecnología utilizada, entre otros aspectos. En resumen, los requerimientos funcionales se centran en lo que el sistema debe hacer, mientras que los requerimientos no funcionales se centran en cómo debe hacerlo y con qué características.

### Respuesta Esperada:

La principal diferencia entre los requerimientos funcionales y no funcionales es que los primeros se refieren a las declaraciones de los servicios que debe proporcionar el sistema y cómo debe comportarse en situaciones particulares, mientras que los segundos se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como limitaciones de tiempo, proceso de desarrollo, rendimiento, calidad, normas, tecnología, etc.

### Contexto Relevante:

1. Clasificación de los requerimientos. Los requerimientos funcionales •Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la for.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos no funcionales técnicas para validación de requerimientos 1 validación de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....
3. presentación de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escritas.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 that talks about validation techniques, and node 3 that discusses a course presentation, are correctly ranked lower than the highly relevant node 1 that clearly explains the difference between functional and non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly explains the difference between functional and non-functional requirements.

Contextual Relevancy: 0.37 (Threshold: 0.5) - The score is 0.37 because the retrieval context mostly lacks information about the difference between functional and non-functional requirements, as stated in the reasons for irrelevancy, and only a few statements, such as 'requerimientos funcionales y no funcionales', are vaguely related to the input.

## Caso de Prueba: 121 - Éxito

### Input del Usuario:

¿Cuál es la principal diferencia entre los requerimientos funcionales y no funcionales en la clasificación de los servicios del sistema?

### Respuesta Generada:

La principal diferencia entre los requerimientos funcionales y no funcionales es que los requerimientos funcionales se refieren a las funciones específicas que debe realizar el sistema, es decir, a los servicios que debe proporcionar el sistema, la forma en que debe reaccionar a las entradas y la forma en que debe comportarse en situaciones particulares (Documento 3 y 4). Por otro lado, los requerimientos no funcionales se refieren a las propiedades emergentes del sistema, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento, y también a las restricciones del sistema, como la capacidad de los dispositivos de entrada-salida y la representación de datos que se utiliza en la interface del sistema (Documento 1 y 5). Estos requerimientos no funcionales imponen limitaciones a las funciones o servicios ofrecidos por el sistema y se centran en la calidad del sistema, como la seguridad, el rendimiento y la interoperabilidad con otros sistemas. En resumen, los requerimientos funcionales se centran en lo que el sistema debe hacer, mientras que los requerimientos no funcionales se centran en cómo debe hacerlo.

### Respuesta Esperada:

La principal diferencia entre los requerimientos funcionales y no funcionales es que los primeros se refieren a las declaraciones de los servicios que debe proporcionar el sistema y su comportamiento en situaciones particulares, mientras que los segundos se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como tiempo, rendimiento, calidad, normas, tecnología, etc.

### Contexto Relevante:

1. Clasificación de los requerimientos. Los requerimientos funcionales •Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la for.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos no funcionales técnicas para validación de requerimientos 1 validación de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....
3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and relevant answer without any irrelevant information.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node, which defines non-functional requirements, and the third-ranked node, which defines functional requirements, are correctly placed above the second-ranked node, which is not directly related to the input question and does not provide relevant information.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context has some relevant statements that explain the difference between functional and non-functional requirements, such as 'requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema' and 'requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos', but most of the statements are irrelevant and don't address the input question.

## Caso de Prueba: 122 - Éxito

### Input del Usuario:

¿Cuál es la principal diferencia entre los requerimientos funcionales y no funcionales en un sistema?

### Respuesta Generada:

Según los documentos proporcionados, la principal diferencia entre los requerimientos funcionales y no funcionales en un sistema es que: \* Los requerimientos funcionales se refieren a las funciones específicas que debe realizar el sistema, es decir, lo que el sistema debe hacer. Estos requerimientos describen el comportamiento del sistema en condiciones específicas y establecen explícitamente lo que el sistema debe hacer y, en algunos casos, lo que no debe hacer. (Documento 3 y 5) \* Los requerimientos no funcionales, por otro lado, se refieren a las propiedades emergentes del sistema, como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, la representación de datos en la interfaz del sistema, entre otros. Estos requerimientos imponen restricciones que limitan los requerimientos funcionales y se refieren a cómo debe comportarse el sistema, más que a qué debe hacer. (Documento 1, 2 y 4) En resumen, los requerimientos funcionales se centran en qué debe hacer el sistema, mientras que los requerimientos no funcionales se centran en cómo debe hacerlo.

### Respuesta Esperada:

La principal diferencia entre los requerimientos funcionales y no funcionales en un sistema es que los primeros se refieren a las declaraciones de los servicios que debe proporcionar el sistema y su comportamiento en situaciones particulares, mientras que los segundos se refieren a las limitaciones en los servicios o funciones ofrecidas por el sistema, como de tiempo, rendimiento, calidad, normas, tecnología, etc.

### Contexto Relevante:

1. Clasificación de los requerimientos. Los requerimientos funcionales •Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la for.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....
3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the top-ranked node providing a clear distinction between non-functional and functional requirements, the second node further clarifying the difference, and the third node directly addressing the definition of functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which explicitly mentions the key differences between functional and non-functional requirements.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the input, providing a comprehensive explanation of the differences between functional and non-functional requirements in a system.

## Caso de Prueba: 123 - Fallo

### Input del Usuario:

¿Cuál es la categoría de requerimientos que se refiere a las limitaciones en los servicios o funciones ofrecidas por el sistema, como tiempo, proceso de desarrollo, rendimiento, calidad, normas, tecnología, etc.?

### Respuesta Generada:

Según los documentos proporcionados, la categoría de requerimientos que se refiere a las limitaciones en los servicios o funciones ofrecidas por el sistema, como tiempo, proceso de desarrollo, rendimiento, calidad, normas, tecnología, etc. se llama "Requerimientos no funcionales". En específico, según el Documento 1, los requerimientos no funcionales se definen como "limitaciones en los servicios o funciones ofrecidas por el sistema como de tiempo, limitaciones en el proceso de desarrollo, rendimiento de calidad, normas tecnología, etc." Además, el Documento 2 proporciona una definición más detallada de los requerimientos no funcionales, indicando que se refieren a "las propiedades emergentes del sistema, como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, etc." y que también definen "las restricciones del sistema, como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interfaz del sistema". Por lo tanto, la respuesta a la pregunta es "Requerimientos no funcionales".

### Respuesta Esperada:

La categoría de requerimientos que se refiere a las limitaciones en los servicios o funciones ofrecidas por el sistema, como tiempo, proceso de desarrollo, rendimiento, calidad, normas, tecnología, etc. es la de los requerimientos no funcionales.

### Contexto Relevante:

1. Clasificación de los requerimientos. Los requerimientos funcionales •Declaraciones de los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la for.....

### Contexto Recuperado:

1. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which discusses 'requerimientos funcionales...' and is not related to non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st and 2nd nodes in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.45 (Threshold: 0.5) - The score is 0.45 because the retrieval context is mostly irrelevant to the input due to the mismatch between the input's focus on non-functional requirements and the context's discussion of functional requirements, as stated in the reasons for irrelevancy: 'The input is asking about non-functional requirements, but this statement is about functional requirements.' and 'The statement does not mention limitations or constraints, which is the main topic of the input.'



## Caso de Prueba: 124 - Fallo

### Input del Usuario:

¿Qué tipo de información describen los requisitos funcionales sobre el sistema?

### Respuesta Generada:

Según los documentos proporcionados, los requisitos funcionales describen el comportamiento del sistema en condiciones específicas. Específicamente, describen los servicios que el sistema debe proporcionar, la forma en que el sistema debe reaccionar a las entradas y la forma en que el sistema debe comportarse en situaciones particulares. En otras palabras, los requisitos funcionales describen lo que el sistema debe hacer, cómo debe reaccionar a determinados insumos y cómo debe comportarse en diferentes situaciones. Esto se refleja en ejemplos de requisitos funcionales como "el sistema debe permitir crear un usuario del sistema" o "el sistema debe permitir registrar participantes a un evento". En resumen, los requisitos funcionales describen el comportamiento y las funcionalidades que se espera que tenga el sistema.

### Respuesta Esperada:

Los requisitos funcionales describen el comportamiento del sistema en condiciones específicas, es decir, cómo reaccionará a determinados insumos y qué servicios prestará.

### Contexto Relevante:

1. Requerimientos funcionales. Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Generalmente,.....

### Contexto Recuperado:

1. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....

3. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....  
.....

### Métricas de Evaluación:

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the output mostly addresses the input question, but includes some unnecessary details about specific functionalities of the system.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts correctly describes functional requirements, and the third-ranked node also provides relevant information, but the second-ranked node is an irrelevant node about 'requerimientos no funcionales', which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly matches the expected output, with the 1st node in the retrieval context accurately describing the system's behavior in specific conditions.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context mostly talks about non-functional requirements, time constraints, development process constraints, quality, and performance, which are not directly related to the input about functional requirements, as stated in reasons for irrelevancy.

## Caso de Prueba: 125 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de los requisitos funcionales en el diseño de un sistema?

### Respuesta Generada:

Según el Documento 1, los requisitos funcionales tienen como propósito principal describir los servicios que prestará el sistema y cómo reaccionará a determinados insumos. En otras palabras, los requisitos funcionales definen el comportamiento del sistema en condiciones específicas. Además, según el Documento 5, los requisitos funcionales declaran los servicios que debe proporcionar el sistema, la forma en que el sistema debe reaccionar a las entradas y la forma en que el sistema debe comportarse en situaciones particulares. En resumen, el propósito principal de los requisitos funcionales es definir las características y comportamientos de un sistema en términos de sus servicios y reacciones a las entradas, lo que permite a los diseñadores y desarrolladores crear un sistema que se ajuste a los objetivos y necesidades del cliente.

### Respuesta Esperada:

El propósito principal de los requisitos funcionales en el diseño de un sistema es describir el comportamiento del sistema en condiciones específicas, es decir, definir los servicios que prestará el sistema y cómo reaccionará a determinados insumos.

### Contexto Relevante:

1. Requerimientos funcionales. Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Generalmente,.....

### Contexto Recuperado:

1. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

2. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....

3. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1 which clearly defines the purpose of functional requirements, are ranked higher than the irrelevant nodes, like node 2 which only provides examples, and node 3 which discusses non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which matches the description of the system's behavior and services.

Contextual Relevancy: 0.77 (Threshold: 0.5) - The score is 0.77 because the retrieval context provides relevant statements that directly address the main purpose of functional requirements, such as 'requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos' and 'los requerimientos funcionales describen el comportamiento del sistema en condiciones especificas', despite some irrelevant statements.

## Caso de Prueba: 126 - Fallo

### Input del Usuario:

¿Qué aspecto del sistema se describe en los requerimientos funcionales?

### Respuesta Generada:

Según los documentos proporcionados, los requerimientos funcionales describen el comportamiento del sistema en condiciones específicas. Específicamente, los requerimientos funcionales declaran los servicios que el sistema debe proporcionar, la forma en que el sistema debe reaccionar a las entradas y la forma en que el sistema debe comportarse en situaciones particulares. Por lo tanto, el aspecto del sistema que se describe en los requerimientos funcionales es el comportamiento del sistema en diferentes situaciones, es decir, cómo el sistema va a reaccionar y responder a diferentes inputs y condiciones.

### Respuesta Esperada:

El comportamiento del sistema en condiciones específicas.

### Contexto Relevante:

1. Requerimientos funcionales. Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Generalmente,.....

### Contexto Recuperado:

1. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....
2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
3. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....  
.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a direct and relevant response!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts directly answers the question, and the third-ranked node provides a similar description, but the second-ranked node, which is an irrelevant node, should be ranked lower as it discusses a different concept, 'requerimientos no funcionales', which does not address the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly matches the expected output, with the 1st node in the retrieval context directly stating 'el comportamiento del sistema en condiciones especificas'.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context is mostly irrelevant to the input, as it primarily discusses non-functional requirements, with only a few statements mentioning functional requirements, such as 'los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema' and 'los requerimientos funcionales describen el comportamiento del sistema en condiciones especificas'.

## Caso de Prueba: 127 - Fallo

### Input del Usuario:

¿Qué tipo de información se utiliza para describir el comportamiento del sistema en condiciones específicas?

### Respuesta Generada:

Según el Documento 2, los Requerimientos Funcionales son declaraciones de los servicios que prestará el sistema, y describen el comportamiento del sistema en condiciones específicas.

### Respuesta Esperada:

Requerimientos funcionales.

### Contexto Relevante:

1. Requerimientos funcionales. Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Generalmente,.....

### Contexto Recuperado:

1. historia de usuario epicas hu como usuario administrador del sistema deseo poder gestionar usuarios para poder administrar los usuarios esta epica se puede subdividir en como usuario administrador des.....
2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....
3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output provides some relevant information about functional requirements, but fails to directly address the specific question about the type of information used to describe system behavior.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first node in the retrieval context, which is about user stories, is ranked higher than the second node, which directly answers the question about describing system behavior, and the third node, which is about specifying requirements, is also ranked higher than the second node.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the sentence 'Requerimientos funcionales.' does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.86 (Threshold: 0.5) - The score is 0.86 because the retrieval context is mostly relevant, except for a few statements that mention 'declaraciones de los servicios' and 'lo que el sistema no debe hacer', which are not directly related to describing the behavior of the system in specific conditions, but the majority of the statements, such as 'los requerimientos funcionales describen el comportamiento del sistema en condiciones especificas', are highly relevant to the input.



## Caso de Prueba: 128 - Éxito

### Input del Usuario:

¿Qué describen los requerimientos funcionales en relación con el comportamiento del sistema?

### Respuesta Generada:

Según el Documento 1, los requerimientos funcionales describen el comportamiento del sistema en condiciones específicas, es decir, describen la forma en que el sistema reaccionará a determinados insumos. Esto significa que los requerimientos funcionales establecen cómo el sistema debe comportarse en situaciones particulares. Además, según el Documento 7, los requerimientos funcionales se suelen especificar en lenguaje natural y se expresan de forma individual, organizándose de forma jerárquica a distintos niveles de detalle. Esto indica que los requerimientos funcionales describen de manera clara y detallada la forma en que el sistema debe comportarse en diferentes situaciones. En resumen, los requerimientos funcionales describen la forma en que el sistema debe reaccionar a determinados insumos y se comportar en condiciones específicas, estableciendo cómo el sistema debe funcionar en diferentes situaciones.

### Respuesta Esperada:

Los requerimientos funcionales describen el comportamiento del sistema en condiciones específicas.

### Contexto Relevante:

1. Requerimientos funcionales. Los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Generalmente,.....

### Contexto Recuperado:

1. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....
2. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....  
.....
3. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....

### Métricas de Evaluación:

Answer Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because the output partially addresses the question by discussing functional requirements, but gets sidetracked by explaining how they are specified, expressed, and organized, rather than focusing on what they describe in relation to system behavior.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1, are ranked higher than irrelevant nodes, like node 2 and node 3, which should be ranked lower since they don't directly address the question or provide information about the description of functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly matches the 1st node in the retrieval context, resulting in a flawless recall.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context is mostly relevant, with statements like 'requerimientos funcionales describen el comportamiento del sistema en condiciones especificas' and 'los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos' directly addressing the input, but is brought down by some irrelevant information, such as the mention of 'requerimientos no funcionales'.

## Caso de Prueba: 129 - Fallo

### Input del Usuario:

¿Cuál es la característica principal de la forma en que se expresan los requerimientos funcionales en lenguaje natural?

### Respuesta Generada:

Según el Documento 1, la característica principal de la forma en que se expresan los requerimientos funcionales en lenguaje natural es que se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle.

### Respuesta Esperada:

La característica principal de la forma en que se expresan los requerimientos funcionales en lenguaje natural es que se suelen especificar en lenguaje natural.

### Contexto Relevante:

1. Requerimientos funcionales. Los requerimientos. se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica (a distintos niveles de detalle), a me.....

### Contexto Recuperado:

1. requerimientos funcionales los requerimientos se suelen especificar en lenguaje natural se expresan de forma individual y se organizan de forma jerarquica a distintos niveles de detalle a menudo se nu.....

2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

3. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output partially addresses the input question, but gets sidetracked by describing the organization of functional requirements, which is not the main focus of the question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly answering the question and the subsequent nodes being irrelevant, such as the second node providing general information about functional requirements and the third node discussing non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in retrieval context, which explicitly mentions the key phrase, resulting in a flawless match.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because the retrieval context is mostly irrelevant, with multiple statements talking about non-functional requirements, as stated in the reasons for irrelevancy, whereas only a few statements like 'requerimientos funcionales los requerimientos se suelen especificar en lenguaje natural' and 'los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos' are actually relevant to the input.

## Caso de Prueba: 130 - Éxito

### Input del Usuario:

¿Cuál es el propósito de organizar los requerimientos funcionales de forma jerárquica?

### Respuesta Generada:

Según el Documento 1, el propósito de organizar los requerimientos funcionales de forma jerárquica es facilitar su gestión. Esto permite organizar los requerimientos de forma clara y estructurada, a diferentes niveles de detalle, lo que ayuda a entender y manejar mejor los requisitos del sistema.

### Respuesta Esperada:

El propósito de organizar los requerimientos funcionales de forma jerárquica es para expresarlos a distintos niveles de detalle.

### Contexto Relevante:

1. Requerimientos funcionales. Los requerimientos. se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica (a distintos niveles de detalle), a me.....

### Contexto Recuperado:

1. requerimientos funcionales los requerimientos se suelen especificar en lenguaje natural se expresan de forma individual y se organizan de forma jerarquica a distintos niveles de detalle a menudo se nu.....

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which talks about non-functional requirements, and node 3 which does not explicitly mention the purpose of organizing functional requirements hierarchically, are correctly ranked lower than the highly relevant node 1 that directly answers the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions the same concepts and phrases, resulting in a flawless match.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context mostly talks about non-functional requirements, which is not related to the purpose of organizing functional requirements in a hierarchical way, as stated in the reasons for irrelevancy, but it does contain some relevant statements about functional requirements being organized hierarchically and describing system behavior.

## Caso de Prueba: 131 - Fallo

### Input del Usuario:

¿Cuál es la ventaja de numerar los requerimientos funcionales en un proyecto?

### Respuesta Generada:

Según el Documento 3, la ventaja de numerar los requerimientos funcionales en un proyecto es para facilitar su gestión. Al asignar un número único a cada requisito funcional, se puede identificar y tracking de manera fácil y clara, lo que permite una mejor organización y seguimiento de los requisitos del proyecto.

### Respuesta Esperada:

La ventaja de numerar los requerimientos funcionales en un proyecto es facilitar su gestión.

### Contexto Relevante:

1. Requerimientos funcionales. Los requerimientos. se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica (a distintos niveles de detalle), a me.....

### Contexto Recuperado:

1. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....

2. requerimientos no funcionales técnicas para obtener requerimientos la entrevista es un diálogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

3. requerimientos funcionales los requerimientos se suelen especificar en lenguaje natural se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle a menudo se nu.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, and it's ranked third, with the first and second nodes being irrelevant due to being about 'ejemplos de requerimientos funcionales...' and 'requerimientos no funcionales...' respectively, which are not directly related to the advantage of numbering functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 3rd node in retrieval context, which mentions a similar phrase, ensuring a seamless recall.

Contextual Relevancy: 0.11 (Threshold: 0.5) - The score is 0.11 because the retrieval context mostly contains irrelevant statements that do not provide any advantage of numbering functional requirements, as stated in the reasons for irrelevancy, with only two relevant statements that mention numbering functional requirements to facilitate their management.



## Caso de Prueba: 132 - Éxito

### Input del Usuario:

¿Cuál es el propósito de numerar los requerimientos funcionales en un proyecto?

### Respuesta Generada:

Según el Documento 1, "los requerimientos funcionales se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle. A menudo se numeran para facilitar su gestión". Por lo tanto, el propósito de numerar los requerimientos funcionales es facilitar su gestión. Al asignar un número único a cada requerimiento funcional, se puede identificar y rastrear fácilmente cada uno de ellos a lo largo del proyecto, lo que simplifica la organización y el seguimiento de los requisitos del sistema.

### Respuesta Esperada:

El propósito de numerar los requerimientos funcionales en un proyecto es facilitar su gestión.

### Contexto Relevante:

1. Requerimientos funcionales. Los requerimientos se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica (a distintos niveles de detalle), a menudo se numeran para facilitar su gestión.....

### Contexto Recuperado:

1. requerimientos funcionales los requerimientos se suelen especificar en lenguaje natural se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle a menudo se numeran.....
2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
3. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema debe.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output partially addresses the input question, but lacks a clear explanation of the purpose of numbering functional requirements, instead focusing on how they are specified, expressed, and organized.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which talks about non-functional requirements and node 3 which provides examples of functional requirements, are correctly ranked lower than the relevant node 1, which clearly explains the purpose of numbering functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in retrieval context, which explicitly mentions the purpose of numbering functional requirements.

Contextual Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because the retrieval context is mostly irrelevant, with statements discussing non-functional requirements and unrelated topics, but it does contain some relevant information, such as 'a menudo se numeran para facilitar su gestion', which is related to the purpose of numbering functional requirements in a project.

## Caso de Prueba: 133 - Éxito

### Input del Usuario:

¿Cuáles son los pasos clave para identificar y documentar los requerimientos funcionales en una metodología tradicional de desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, los pasos clave para identificar y documentar los requerimientos funcionales en una metodología tradicional de desarrollo de software son: 1. **Realizar una descripción del requerimiento**: Se debe hacer una descripción del requerimiento lo más precisa posible, utilizando un lenguaje consistente y evitando el uso de jerga informática. 2. **Realizar la especificación de los detalles del requerimiento**: Se debe especificar los detalles del requerimiento necesarios para su implementación. 3. **Identificar los actores y funcionalidades**: Es necesario identificar a los actores clave y las funcionalidades que se requieren en el sistema. 4. **Realizar el análisis de los requerimientos**: Se debe analizar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para el sistema de software. 5. **Elicitar los requerimientos**: Se deben recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. 6. **Realizar una entrevista**: Se pueden realizar entrevistas con los usuarios y partes interesadas para obtener respuestas claras a un conjunto de preguntas planeadas y entender con claridad cada cosa que el cliente desea. 7. **Documentar los requerimientos**: Se deben documentar los requerimientos funcionales de manera clara y detallada, utilizando un formato estándar y asegurando la adherencia al mismo. 8. **Validar los requerimientos**: Se deben validar los requerimientos funcionales para asegurar que se entienden correctamente y se ajustan a los objetivos del proyecto y las necesidades del cliente.

### Respuesta Esperada:

En una metodología tradicional de desarrollo de software, los pasos clave para identificar y documentar los requerimientos funcionales son: 1. Recopilación de requisitos: se recopilan los requisitos funcionales a través de entrevistas con los stakeholders, análisis de documentos y observación de procesos. 2. Análisis de requisitos: se analizan los requisitos recopilados para identificar los requisitos funcionales y no funcionales. 3. Definición de requisitos: se definen los requisitos funcionales en términos de entradas, procesos y salidas. 4. Documentación de requisitos: se documentan los requisitos funcionales en un documento de requisitos, que suele ser el Documento de Requisitos del Sistema (DRS). 5. Revisión y aprobación: se revisa y aprueba el documento de requisitos por parte de los stakeholders. Es importante destacar que la documentación de los requisitos funcionales es fundamental para asegurar que se cumplan las necesidades del cliente y se desarrollen sistemas que satisfagan sus expectativas.

### Contexto Relevante:

1. Requerimientos funcionales. Requerimientos en las metodologías tradicionales...

### Contexto Recuperado:

1. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....
2. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node explaining the step-by-step approach, the second node discussing techniques for obtaining requirements, and the third node detailing the process of specifying requirements, all of which perfectly align with the input's question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 3rd node in the retrieval context, which comprehensively covers the entire software development process, from identifying user needs to reviewing and approving the document of requirements.

Contextual Relevancy: 0.92 (Threshold: 0.5) - The score is 0.92 because the retrieval context provides specific and relevant steps for identifying and documenting functional requirements in a traditional software development methodology, such as 'un requerimiento se especifica en dos pasos' and 'elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software', which directly address the input question.

## Caso de Prueba: 134 - Éxito

### Input del Usuario:

¿Cuál es el enfoque principal para la recopilación de Requerimientos en las metodologías tradicionales como el modelo cascada?

### Respuesta Generada:

Según el Documento 5, en las metodologías tradicionales, como el modelo cascada, el enfoque principal para la recopilación de Requerimientos es la Elaboración, que se basa en actividades como la Obtención de Requerimientos, Identificación de Actores, Identificación de Funcionalidades, personas, otro sistema, dispositivos, lo que se desea que el sistema haga. En otras palabras, en las metodologías tradicionales, se busca recopilar y especificar los Requerimientos Funcionales del sistema de manera exhaustiva y detallada, con la participación de los actores involucrados y la identificación de las funcionalidades y necesidades del sistema. Además, según el Documento 6, en las metodologías tradicionales, los Requerimientos Funcionales se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle, lo que permite una mejor gestión y comprensión de los Requerimientos del sistema.

### Respuesta Esperada:

El enfoque principal para la recopilación de Requerimientos en las metodologías tradicionales como el modelo cascada es la recopilación de Requerimientos funcionales.

### Contexto Relevante:

1. Requerimientos funcionales. Requerimientos en las metodologías tradicionales...

### Contexto Recuperado:

1. ejemplos de requerimientos funcionales en las metodologias tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....

2. requerimientos funcionales requerimientos en las metodologias tradicionales...

3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 0.77 (Threshold: 0.5) - The score is 0.77 because while the output addresses the main focus for requirement gathering in traditional methodologies, it is brought down by mentions of irrelevant concepts like personas, otro sistema, and dispositivos.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which talks about agile

methodologies and principles, and is not relevant to the topic of requirement gathering in traditional methodologies like the waterfall model.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, as the sentence explicitly mentions 'recopilación de Requerimientos funcionales' which is directly attributed to the 2nd node in the retrieval context.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context has some relevant statements mentioning traditional methodologies, such as 'ejemplos de requerimientos funcionales en las metodologías tradicionales' and 'requerimientos en las metodologías tradicionales', but is largely overshadowed by irrelevant statements that focus on agile processes and teams, as stated in the reasons for irrelevancy.

## Caso de Prueba: 135 - Fallo

### Input del Usuario:

¿Cómo puedo asegurarme de que los requisitos sean lo suficientemente claros y detallados para evitar malentendidos y errores en el desarrollo del proyecto?

### Respuesta Generada:

Para asegurarte de que los requisitos sean lo suficientemente claros y detallados, existen varias prácticas que puedes implementar: 1. **\*\*Divide los requisitos en tareas sencillas\*\***: Según el Documento 1, es recomendable dividir una historia de usuario en tareas sencillas para asegurarte de que sean manejables y claras. 2. **\*\*Define los requisitos de manera clara y concisa\*\***: Asegúrate de que los requisitos estén redactados de tal manera que sean comprensibles para usuarios sin conocimientos técnicos avanzados, como se menciona en el Documento 3. 3. **\*\*Incluye archivos y documentación de apoyo\*\***: Si es necesario, incluye archivos y documentación de apoyo para proporcionar más contexto y claridad a los requisitos, como se sugiere en el Documento 1. 4. **\*\*Utiliza técnicas de validación de requerimientos\*\***: Utiliza técnicas como la validación de expertos, prototipado de interfaz de usuario, y recorrido de BPM mencionadas en el Documento 5 para asegurarte de que los requisitos sean claros y detallados. 5. **\*\*Realiza el Definition of Ready (DoR)\*\***: Antes de iniciar un sprint, asegúrate de que los elementos del Product Backlog cumplan con los criterios del DoR, como se menciona en el Documento 4, para garantizar que los requisitos estén bien definidos y claros. 6. **\*\*Trabaja en estrecha colaboración con los interesados\*\***: Asegúrate de trabajar en estrecha colaboración con los usuarios y otros interesados para asegurarte de que los requisitos sean claros y detallados, como se menciona en el Documento 6. 7. **\*\*Utiliza lenguaje natural y jerarquía de requisitos\*\***: Utiliza lenguaje natural y organiza los requisitos de manera jerárquica, como se sugiere en el Documento 10, para facilitar la comprensión y gestión de los requisitos. Al implementar estas prácticas, podrás asegurarte de que los requisitos sean lo suficientemente claros y detallados para evitar malentendidos y errores en el desarrollo del proyecto.

### Respuesta Esperada:

Para asegurarte de que los requisitos sean lo suficientemente claros y detallados, es importante implementar las siguientes prácticas: 1. Establecer un proceso de recopilación de requisitos estructurado, que incluya entrevistas con los stakeholders, análisis de documentos y observación de procesos. 2. Utilizar herramientas de modelado de requisitos, como diagramas de flujo o casos de uso, para visualizar y clarificar los requisitos. 3. Establecer un lenguaje común y evitar términos ambiguos o vagos en la documentación de requisitos. 4. Realizar revisiones y aprobaciones periódicas de los requisitos con los stakeholders para asegurarse de que se entienden correctamente. 5. Proporcionar ejemplos concretos y escenarios de uso para ilustrar los requisitos y reducir la ambigüedad. 6. Establecer un proceso de gestión de cambios para manejar cualquier modificación en los requisitos durante el desarrollo del proyecto. Al implementar estas prácticas, podrás reducir la imprecisión en los requerimientos y minimizar el riesgo de malentendidos y errores en el desarrollo del proyecto.

### Contexto Relevante:

1. Imprecisión en los requerimientos....

### Contexto Recuperado:

1. características de la historia de usuario ser lo suficientemente completa como para demostrar el valor para el usuario centrarse en el usuario si no se identifica necesidades precisas se puede iniciar.....
2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....
3. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the response perfectly addresses the input question, providing clear and detailed information to ensure requirements are well-defined and avoid misunderstandings and errors in project development.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 in the retrieval contexts provides guidance on creating clear and concise user stories, and node 3 highlights the importance of precise requirements, but node 2 discussing non-functional requirements should be ranked lower as it's not directly related to ensuring clear requirements in the context of the question.

Contextual Recall: 0.14 (Threshold: 0.5) - The score is 0.14 because only sentence 1 in the expected output can be attributed to the 3rd node in retrieval context, which mentions 'imprecision en los requerimientos', while the rest of the sentences lack matching information in the retrieval context.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context provides multiple relevant statements that directly address the input question, such as 'ser lo suficientemente completa como para demostrar el valor para el usuario', 'ser breve sencilla y clara', and 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision', which all emphasize the importance of clear and detailed requirements to avoid misunderstandings and errors.



## Caso de Prueba: 136 - Fallo

### Input del Usuario:

¿Cómo se pueden identificar y abordar las imprecisiones en los requisitos de un proyecto para evitar malentendidos y errores en la implementación?

### Respuesta Generada:

Según el contexto proporcionado, se pueden identificar y abordar las imprecisiones en los requisitos de un proyecto de la siguiente manera: 1. **\*\*Elicitación de requisitos\*\***: Se deben realizar entrevistas con los usuarios y partes interesadas para obtener información clara y precisa sobre los requisitos del proyecto. Esta técnica se menciona en el Documento 3. 2. **\*\*Análisis de requisitos\*\***: Se deben analizar los requisitos identificados para asegurarse de que sean claros, precisos y no ambiguos. Esto se puede lograr mediante la creación de historias de usuario, como se menciona en el Documento 9. 3. **\*\*Validación de requisitos\*\***: Se deben validar los requisitos mediante técnicas como la validación de expertos, prototipado de interfaz de usuario y recorrido de BPM. Estas técnicas se mencionan en los Documentos 5 y 6. 4. **\*\*Especificar requisitos\*\***: Se deben especificar los requisitos de manera clara y precisa, utilizando un lenguaje natural y organizándolos de forma jerárquica. Esto se menciona en el Documento 10. 5. **\*\*Revisión y retroalimentación\*\***: Se debe revisar y obtener retroalimentación de los usuarios y partes interesadas sobre los requisitos identificados y especificados. Esto se puede lograr mediante la realización de retrospectivas, como se menciona en el Documento 2. Al seguir estos pasos, se puede identificar y abordar las imprecisiones en los requisitos de un proyecto, lo que ayuda a evitar malentendidos y errores en la implementación.

### Respuesta Esperada:

Para identificar y abordar las imprecisiones en los requisitos de un proyecto, se pueden seguir los siguientes pasos: 1. Revisar y analizar cuidadosamente los requisitos: Es importante leer y analizar los requisitos con detenimiento para detectar cualquier ambigüedad o falta de claridad. 2. Preguntar y aclarar: Si hay alguna duda o inquietud sobre un requisito, es fundamental preguntar y aclarar con el cliente o stakeholder para evitar malentendidos. 3. Establecer un lenguaje común: Es importante establecer un lenguaje común y evitar el uso de términos ambiguos o vagos para describir los requisitos. 4. Definir métricas y criterios de éxito: Es fundamental definir métricas y criterios de éxito claros para cada requisito para asegurarse de que se cumplan los objetivos del proyecto. 5. Realizar pruebas y prototipos: Realizar pruebas y prototipos puede ayudar a identificar imprecisiones en los requisitos y aclarar cualquier duda o inquietud. 6. Revisar y actualizar los requisitos: Es importante revisar y actualizar los requisitos regularmente para asegurarse de que se ajusten a los cambios y necesidades del proyecto. Siguiendo estos pasos, se puede reducir el riesgo de malentendidos y errores en la implementación del proyecto.

### Contexto Relevante:

1. Imprecisión en los requerimientos....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por

los desarrollado.....

2. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

3. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant response to the question asked!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because only one irrelevant node, ranked 2nd, discussing 'practicas agiles en scrum' and 'retrospectiva', is ranked higher than a relevant node, ranked 3rd, mentioning 'entrevista' as a technique to obtain requirements, which should be ranked higher.

Contextual Recall: 0.29 (Threshold: 0.5) - The score is 0.29 because only sentences 1 and 2 in the expected output can be attributed to node 1 in the retrieval context, while the rest of the sentences lack connection to the context.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context is mostly discussing agile practices and Scrum retrospective, which has no relation to identifying and addressing inaccuracies in project requirements, as stated in the reasons for irrelevancy, such as 'The statement is talking about agile practices in Scrum retrospective, which has no relation to identifying and addressing inaccuracies in project requirements.'

## Caso de Prueba: 137 - Éxito

### Input del Usuario:

¿Cuáles son las técnicas más efectivas para identificar y abordar la imprecisión en los requisitos del sistema durante el análisis de requisitos?

### Respuesta Generada:

Según los documentos proporcionados, las técnicas más efectivas para identificar y abordar la imprecisión en los requisitos del sistema durante el análisis de requisitos son: 1. **\*\*Entrevistas con los usuarios y partes interesadas\*\***: Estas permiten recopilar información directamente de los usuarios y partes interesadas, lo que ayuda a identificar y aclarar requisitos ambiguos o imprecisos (Documento 1). 2. **\*\*Estudios de sistemas parecidos\*\***: Analizar sistemas similares puede ayudar a identificar posibles requisitos que no han sido considerados y a abordar la imprecisión en los requisitos (Documento 3). 3. **\*\*Talleres y lluvia de ideas\*\***: Estas técnicas facilitan la comunicación entre los usuarios y los desarrolladores, lo que ayuda a identificar y aclarar requisitos imprecisos y a alcanzar un consenso sobre los requisitos del sistema (Documento 3). 4. **\*\*Prototipos y modelado de procesos de negocio\*\***: Crear prototipos y modelos de procesos de negocio puede ayudar a identificar y abordar la imprecisión en los requisitos, ya que permiten probar y refinar los requisitos del sistema de manera iterativa (Documentos 4 y 3). 5. **\*\*Análisis de requisitos\*\***: Realizar un análisis detallado de los requisitos del sistema, identificando y clasificando los requisitos funcionales y no funcionales, puede ayudar a identificar y abordar la imprecisión en los requisitos (Documentos 7, 8 y 9). 6. **\*\*Revisión y validación de requisitos\*\***: Revisar y validar los requisitos del sistema con los usuarios y partes interesadas puede ayudar a asegurarse de que los requisitos sean correctos y no imprecisos (Documentos 3, 7 y 10). En resumen, estas técnicas permiten recopilar información precisa y clara sobre los requisitos del sistema, identificar y abordar la imprecisión en los requisitos, y asegurarse de que los requisitos sean claros y comprensibles para todos los involucrados.

### Respuesta Esperada:

Basándome en el contexto proporcionado, algunas de las técnicas más efectivas para identificar y abordar la imprecisión en los requisitos del sistema durante el análisis de requisitos son: \* Análisis de requisitos iterativos: Realizar un análisis de requisitos en varias iteraciones, cada una con un nivel de detalle cada vez más fino, para identificar y refinar los requisitos. \* Técnicas de elicitation: Utilizar técnicas como entrevistas, grupos de enfoque, observación y prototipado para recopilar información de los stakeholders y clarificar los requisitos. \* Análisis de escenarios: Identificar y analizar los escenarios de uso del sistema para entender mejor los requisitos y detectar posibles imprecisiones. \* Revisión cruzada: Realizar una revisión cruzada de los requisitos con los stakeholders y otros miembros del equipo para detectar y abordar las imprecisiones. \* Modelado de requisitos: Utilizar modelos de requisitos, como diagramas de flujo de datos o modelos de entidad-relación, para visualizar y analizar los requisitos y detectar posibles imprecisiones. Es importante destacar que la imprecisión en los requisitos puede ser causada por una variedad de factores, como la falta de comunicación, la ambigüedad en la definición de los requisitos o la falta de experiencia en el dominio del problema.

### Contexto Relevante:

1. Imprecisión en los requerimientos....

## **Contexto Recuperado:**

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
2. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....
3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant response without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node highlighting the importance of 'la entrevista' for eliciting requirements, the second node emphasizing the need for clear requirements, and the third node providing a comprehensive overview of the requirement specification process.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the retrieval context, with each sentence attributed to a specific node, such as the 1st node for sentence 2, the 2nd node for sentences 1, 4, and 6, and the 3rd node for sentence 5, demonstrating a flawless contextual understanding.

Contextual Relevancy: 0.93 (Threshold: 0.5) - The score is 0.93 because the retrieval context provides specific and relevant techniques for identifying and addressing imprecision in system requirements, such as conducting interviews, asking clear questions, and studying the problem, which directly address the input question.

## Caso de Prueba: 138 - Fallo

### Input del Usuario:

Cómo se pueden identificar y abordar las ambigüedades en los requisitos del proyecto para evitar malentendidos y asegurar que se cumplan los objetivos esperados?

### Respuesta Generada:

Según los documentos proporcionados, para identificar y abordar las ambigüedades en los requisitos del proyecto, se pueden implementar las siguientes prácticas: 1. **\*\*Elicitación de requisitos\*\***: Se debe realizar una entrevista con los usuarios y partes interesadas para entender claramente cada cosa que el cliente desea. Es importante identificar posibles entrevistados, estudiar el problema planteado, familiarizarse con el vocabulario del negocio y tomar apuntes o grabar la reunión (Documento 3). 2. **\*\*Análisis de requisitos\*\***: Se debe analizar los requisitos para identificar posibles ambigüedades y asegurarse de que sean comprensibles para usuarios sin conocimientos técnicos avanzados (Documento 2). 3. **\*\*Técnicas de validación de requisitos\*\***: Se pueden utilizar técnicas como la validación de expertos, prototipado de interfaz de usuario y recorrido de BPM para asegurarse de que los requisitos sean claros y precisos (Documentos 5 y 9). 4. **\*\*Definición de Done (DoD)\*\***: Se puede establecer un criterio de aceptación claro para cada requisito, asegurándose de que se cumplan todos los criterios de aceptación definidos previamente (Documento 6). 5. **\*\*Revisión y aprobación de requisitos\*\***: Se deben revisar y aprobar los requisitos con los usuarios y partes interesadas para asegurarse de que se entiendan correctamente y se cumplan los objetivos esperados. Al implementar estas prácticas, se pueden identificar y abordar las ambigüedades en los requisitos del proyecto, lo que ayudará a evitar malentendidos y asegurar que se cumplan los objetivos esperados.

### Respuesta Esperada:

Para identificar y abordar las ambigüedades en los requisitos del proyecto, se pueden seguir los siguientes pasos: 1. Revisar cuidadosamente los requisitos: Realizar una revisión detallada de los requisitos para detectar cualquier ambigüedad o imprecisión. 2. Preguntar y aclarar: Hacer preguntas a los stakeholders para aclarar cualquier requisito ambiguo y asegurarse de entender correctamente lo que se espera. 3. Definir términos y conceptos: Establecer una definición clara de los términos y conceptos utilizados en los requisitos para evitar malentendidos. 4. Establecer un lenguaje común: Asegurarse de que todos los miembros del equipo y los stakeholders utilicen un lenguaje común y compartan la misma comprensión de los requisitos. 5. Documentar y aprobar: Documentar los requisitos clarificados y obtener la aprobación de los stakeholders para asegurarse de que todos estén de acuerdo en los objetivos y expectativas del proyecto. Al seguir estos pasos, se puede reducir significativamente el riesgo de malentendidos y asegurar que el proyecto se ajuste a los objetivos esperados.

### Contexto Relevante:

1. Imprecisión en los requerimientos....

### Contexto Recuperado:

1. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de

de.....

2. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollados.....

3. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise response that directly answers the query, without any irrelevant statements!

Contextual Precision: 0.58 (Threshold: 0.5) - The score is 0.58 because the first node, ranked 1, is an irrelevant node as it talks about 'practicas agiles en scrum retrospectiva...' which is not related to identifying and addressing ambiguities in project requirements, and should be ranked lower than the subsequent nodes.

Contextual Recall: 0.67 (Threshold: 0.5) - The score is 0.67 because sentences 1, 3, and 4 in the expected output are well-supported by the 2nd node in the retrieval context, which discusses ambiguous requirements, but sentences 2 and 5 lack clear connections to the retrieval context.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context is mostly irrelevant to the input, with many statements not directly related to identifying and addressing ambiguities in project requirements, as stated in the reasons for irrelevancy, whereas only a few statements, such as 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision' and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios', are relevant to the input.

## Caso de Prueba: 139 - Fallo

### Input del Usuario:

¿Cómo se pueden identificar y abordar las imprecisiones en los requisitos del cliente para evitar malentendidos y retrasos en el proyecto?

### Respuesta Generada:

Según los documentos proporcionados, para identificar y abordar las imprecisiones en los requisitos del cliente, se pueden seguir los siguientes pasos: 1. **\*\*Elicitación\*\***: Realizar entrevistas con los clientes y partes interesadas para recopilar y descubrir los requisitos del sistema de software. Esto se puede hacer a través de diálogos formales o informales, identificando el propósito de la entrevista, preparando preguntas claras y tomando apuntes o grabando la reunión (Documento 4). 2. **\*\*Análisis de requisitos\*\***: Identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. Esto implica analizar las funcionalidades y restricciones del sistema, y determinar qué características debe tener el software (Documento 8). 3. **\*\*Especificación\*\***: Definir qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. Esto se puede lograr a través de la creación de un documento de especificación de requisitos (Documento 8). 4. **\*\*Validación\*\***: Realizar pruebas para asegurar que los requisitos fueron entendidos correctamente, con el fin de garantizar que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente. Esto se puede lograr a través de la validación de expertos, prototipado de interfaz de usuario y recorrido de BPM (Documentos 5 y 7). 5. **\*\*Revisión y retroalimentación\*\***: Realizar revisiones periódicas del proyecto, utilizando prácticas ágiles como Scrum, para analizar cómo se está trabajando y qué problemas se han encontrado. Esto permitirá identificar y abordar las imprecisiones en los requisitos del cliente de manera temprana, y realizar ajustes según sea necesario (Documento 2). 6. **\*\*Comunicación efectiva\*\***: Fomentar la comunicación efectiva entre los responsables de negocio, desarrolladores y clientes, a través de conversaciones cara a cara o virtuales, para asegurar que todos los involucrados tengan una comprensión clara de los requisitos del proyecto (Documento 6). Al seguir estos pasos, se puede reducir el riesgo de malentendidos y retrasos en el proyecto, y asegurar que se cumplan las necesidades y expectativas de los clientes.

### Respuesta Esperada:

Para identificar y abordar las imprecisiones en los requisitos del cliente, se pueden seguir los siguientes pasos: 1. Clarificar y especificar los requisitos: Realizar preguntas detalladas al cliente para aclarar cualquier ambigüedad o vaguedad en los requisitos. 2. Verificar la consistencia: Revisar los requisitos para asegurarse de que sean coherentes y no contradictorios. 3. Establecer un lenguaje común: Utilizar un lenguaje claro y preciso para describir los requisitos, evitando términos ambiguos o técnicos que puedan generar confusión. 4. Documentar los requisitos: Registrar por escrito los requisitos del cliente, incluyendo cualquier suposición o asunción, para evitar malentendidos. 5. Revisar y aprobar los requisitos: Realizar una revisión conjunta con el cliente para asegurarse de que los requisitos sean correctos y aprobarlos formalmente. Al seguir estos pasos, se puede reducir significativamente el riesgo de malentendidos y retrasos en el proyecto, ya que se establece una comprensión clara y precisa de los requisitos del cliente desde el principio.

### Contexto Relevante:

1. Imprecisión en los requerimientos....

### **Contexto Recuperado:**

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....
2. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....
3. detalles de una historia de usuario restricciones 1 solo los usuarios con privilegios de gerente pueden acceder al reporte 2el tiempo de respuesta para loguearse no debe exceder 4 segundos criterios de.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the nodes in the retrieval contexts are perfectly ranked, with the most relevant node, ranked 1, directly addressing the input question, and the irrelevant nodes, ranked 2 and 3, discussing unrelated topics like Scrum retrospectives and user stories.

Contextual Recall: 0.67 (Threshold: 0.5) - The score is 0.67 because the model successfully attributed sentences 1, 3, and 4 to the 1st node in the retrieval context, which emphasizes clear and precise requirements, but struggled to find connections for sentences 2 and 5.

Contextual Relevancy: 0.15 (Threshold: 0.5) - The score is 0.15 because the retrieval context is mostly irrelevant to the input, with many statements discussing Scrum practices, team performance, and other unrelated topics, as stated in the reasons for irrelevancy, such as 'The statement is talking about agile practices in Scrum, which is not related to identifying and addressing client requirement imprecisions.' However, a few relevant statements were found, like 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision', which shows some connection to the input.



## Caso de Prueba: 140 - Éxito

### Input del Usuario:

¿Cómo se pueden evitar las imprecisiones en los requerimientos para asegurar que los desarrolladores entiendan correctamente lo que se necesita?

### Respuesta Generada:

Según los documentos proporcionados, se pueden evitar las imprecisiones en los requerimientos siguiendo las siguientes recomendaciones: 1. Utilizar un formato estándar para todos los requerimientos, lo que ayudará a asegurar la consistencia y claridad en la redacción de los mismos (Documento 3). 2. Utilizar un lenguaje claro y comprensible para los usuarios sin conocimientos técnicos avanzados, evitando el uso de jerga informática (Documento 3). 3. Realizar una descripción del requerimiento lo más precisa posible, y luego especificar los detalles del requerimiento necesarios para su implementación (Documento 3). 4. Identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas a través de la elicitación y análisis de requerimientos (Documento 7). 5. Realizar una especificación detallada de los requerimientos, utilizando un lenguaje natural y organizándolos de forma jerárquica con diferentes niveles de detalle (Documento 6). 6. Involucrar a los usuarios y partes interesadas en el proceso de especificación de requerimientos para asegurar que se entiendan correctamente las necesidades y expectativas (Documento 7). 7. Realizar una validación mediante pruebas para asegurar que los requerimientos fueron entendidos correctamente y que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente (Documento 7). 8. Fomentar la comunicación directa y abierta entre los desarrolladores, usuarios y partes interesadas para reducir la posibilidad de malentendidos y imprecisiones en los requerimientos (Documento 2). Siguiendo estas recomendaciones, se puede reducir significativamente la posibilidad de imprecisiones en los requerimientos, lo que ayudará a asegurar que los desarrolladores entiendan correctamente lo que se necesita y puedan implementar el sistema de software de manera efectiva.

### Respuesta Esperada:

Para evitar imprecisiones en los requerimientos y asegurar que los desarrolladores entiendan correctamente lo que se necesita, es fundamental establecer una comunicación clara y efectiva entre todas las partes involucradas. Esto se puede lograr a través de una definición precisa y detallada de los requerimientos, y asegurándose de que todos los involucrados tengan la misma comprensión de lo que se necesita.

### Contexto Relevante:

1. Imprecisión en los requerimientos Asi fueron definidos los requerimientos Asi lo entendieron los desarrolladores Asi fué resuelto el problema anteriormente Asi fué resuelto el problema ahora Este es e.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. imprecision en los requerimientos asi fueron definidos los requerimientos asi lo entendieron los desarrolladores asi fue resuelto el problema anteriormente asi fue resuelto el problema ahora este es e.....

3. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which explains the importance of precise requirements, is correctly ranked high, and the third node, which provides specific guidelines for writing clear requirements, is also ranked high. However, the second node, ranked 2, is an irrelevant node that focuses on the consequences of imprecise requirements rather than providing a solution, which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in retrieval context, such as node 1 discussing imprecision in requirements and node 3 providing guidelines for clear communication, resulting in a flawless match.

Contextual Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the retrieval context is mostly relevant, with statements like 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision' and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores' directly addressing the input's concern, despite some irrelevant statements.

## Caso de Prueba: 141 - Éxito

### Input del Usuario:

¿Cuáles son las principales causas de la imprecisión en los requerimientos y cómo se pueden evitar en el futuro?

### Respuesta Generada:

Según el contexto proporcionado, la imprecisión en los requerimientos surge cuando estos no son declarados con precisión, lo que puede generar problemas en el desarrollo del sistema. Los requerimientos ambiguos pueden ser interpretados de diferentes maneras por los desarrolladores y usuarios, lo que puede llevar a errores y malentendidos. Las principales causas de la imprecisión en los requerimientos son: 1. Falta de claridad en la redacción de los requerimientos: Los requerimientos deben ser redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados. 2. Falta de precisión en la definición de los requerimientos: Los requerimientos deben ser específicos y detallados, evitando la ambigüedad y la vaguedad. Para evitar la imprecisión en los requerimientos en el futuro, se pueden implementar las siguientes estrategias: 1. Validación de expertos: Personas con experiencia en el desarrollo de sistemas deben revisar y aprobar los requerimientos para asegurarse de que sean precisos y claros. 2. Prototipado de interfaz de usuario: El prototipado de interfaz de usuario permite validar los requerimientos de manera rápida y eficiente, permitiendo identificar y corregir errores y ambigüedades en una etapa temprana del desarrollo. 3. Recorrido de BPM (Business Process Management): El recorrido de BPM permite modelar el proceso del sistema y identificar los requerimientos funcionales y no funcionales de manera clara y precisa. 4. Utilizar historias de usuario: Las historias de usuario permiten describir las necesidades de los usuarios de manera clara y concisa, permitiendo identificar los requerimientos funcionales y no funcionales de manera precisa. 5. Revisión y aprobación de los requerimientos: Es importante que los requerimientos sean revisados y aprobados por todos los involucrados en el desarrollo del sistema, incluyendo a los usuarios y stakeholders. En resumen, la imprecisión en los requerimientos puede llevar a errores y malentendidos en el desarrollo del sistema. Para evitar esto, es importante implementar estrategias de validación y revisión de los requerimientos, como la validación de expertos, prototipado de interfaz de usuario, recorrido de BPM y utilización de historias de usuario.

### Respuesta Esperada:

Según el contexto, las principales causas de la imprecisión en los requerimientos son los problemas de comunicación entre las partes involucradas (ventas, desarrolladores y clientes). Estos problemas llevan a una comprensión diferente de los requerimientos, lo que a su vez conduce a soluciones que no se ajustan a las necesidades reales del cliente. Para evitar la imprecisión en el futuro, es fundamental establecer canales de comunicación claros y efectivos entre todas las partes involucradas, asegurándose de que todos tengan una comprensión común y precisa de los requerimientos.

### Contexto Relevante:

1. Imprecisión en los requerimientos Asi fueron definidos los requerimientos Asi lo entendieron los desarrolladores Asi fué resuelto el problema anteriormente Asi fué resuelto el problema ahora Este es e.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....
2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....
3. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing accurate and relevant information without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which talks about 'requerimientos no funcionales' and validation techniques, and node 3 which explains 'requerimientos no funcionales' and their sources, are correctly ranked lower than the highly relevant node 1, which directly addresses the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, with node 1 in the retrieval context directly addressing communication problems and precise requirement declaration, resulting in a seamless match.

Contextual Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because the retrieval context is only partially relevant, as it includes statements that are not directly related to the causes of imprecision in requirements, such as 'requerimientos no funcionales tecnicas' and 'recorrido de bpm', but also includes relevant statements like 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision' and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores'.

## Caso de Prueba: 142 - Fallo

### Input del Usuario:

¿Cómo se puede evitar la imprecisión en los requerimientos debido a problemas de comunicación entre los clientes, ventas y desarrolladores?

### Respuesta Generada:

Según los documentos proporcionados, para evitar la imprecisión en los requerimientos debido a problemas de comunicación entre los clientes, ventas y desarrolladores, se pueden implementar las siguientes prácticas: 1. **\*\*Requerimientos claros y comprensibles\*\***: Los requerimientos deben ser redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados (Documento 2). 2. **\*\*Participación activa del cliente\*\***: Es importante que el cliente participe activamente en la definición de los requerimientos y en la verificación de que se han cumplido (Documento 3). 3. **\*\*Técnicas para obtener requerimientos\*\***: Se pueden utilizar técnicas como entrevistas con los usuarios (Documento 6) o lluvia de ideas (Documento 9) para obtener requerimientos precisos y evitar malentendidos. 4. **\*\*Especificar requerimientos funcionales y no funcionales\*\***: Es importante especificar tanto los requerimientos funcionales (Documento 4 y 10) como los no funcionales (Documento 5) para entender completamente las necesidades del cliente. 5. **\*\*Revisión y validación\*\***: Es esencial revisar y validar los requerimientos con el cliente y ventas para asegurarse de que se han entendido correctamente (Documento 3). 6. **\*\*Comunicación efectiva\*\***: La comunicación abierta y efectiva entre clientes, ventas y desarrolladores es clave para evitar malentendidos y asegurarse de que se han cumplido los requerimientos (Documento 1). Al implementar estas prácticas, se puede reducir la imprecisión en los requerimientos y asegurarse de que se entregue un producto que se ajuste a las necesidades y expectativas del cliente.

### Respuesta Esperada:

Para evitar la imprecisión en los requerimientos debido a problemas de comunicación entre los clientes, ventas y desarrolladores, es fundamental establecer un canal de comunicación claro y efectivo entre todas las partes involucradas. Esto se puede lograr a través de reuniones periódicas, definiciones claras de los requerimientos y la creación de un documento de requisitos que sea compartido y entendido por todos. De esta manera, se puede asegurar que todos estén en la misma página y que los requerimientos sean entendidos correctamente, evitando así la imprecisión y los problemas que surgen de ella.

### Contexto Relevante:

1. Imprecisión en los requerimientos Asi fueron definidos los requerimientos Asi lo entendieron los desarrolladores Asi fué resuelto el problema anteriormente Asi fué resuelto el problema ahora Este es e.....

### Contexto Recuperado:

1. imprecision en los requerimientos asi fueron definidos los requerimientos asi lo entendieron los desarrolladores asi fue resuelto el problema anteriormente asi fue resuelto el problema ahora este es e.....

2. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the node at rank 1 in the retrieval contexts does not provide a solution to avoid imprecision in requirements, and the node at rank 3 is not directly related to the topic, whereas the node at rank 2 provides a clear method to avoid imprecision, which should be ranked higher.

Contextual Recall: 0.75 (Threshold: 0.5) - The score is 0.75 because the nodes in retrieval context effectively addressed the main topic of imprecision in requirements and its consequences, but lacked direct connection to specific solutions like periodic meetings and shared documents.

Contextual Relevancy: 0.35 (Threshold: 0.5) - The score is 0.35 because the retrieval context has many irrelevant statements that do not address avoiding imprecision in requirements due to communication problems, such as discussing agile practices, team work, and achieving objectives, as stated in the reasons for irrelevancy, whereas only a few statements like 'imprecision en los requerimientos', 'problemas de comunicacion', and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores' are relevant to the input.

## Caso de Prueba: 143 - Fallo

### Input del Usuario:

¿Cómo se puede evitar la imprecisión en los requerimientos y mejorar la comprensión entre los stakeholders para desarrollar soluciones efectivas?

### Respuesta Generada:

Para evitar la imprecisión en los requerimientos y mejorar la comprensión entre los stakeholders para desarrollar soluciones efectivas, se pueden implementar las siguientes estrategias: 1. **\*\*Redacción clara y concisa\*\***: Los requerimientos deben ser redactados de forma clara, concisa y comprensible para todos los stakeholders, incluyendo a los usuarios sin conocimientos técnicos avanzados. 2. **\*\*Técnicas de obtención de requerimientos\*\***: Utilizar técnicas como entrevistas con los usuarios, lluvia de ideas, prototipado de interfaz de usuario, y recorrido de BPM para obtener requerimientos no funcionales y funcionales. 3. **\*\*Validación de expertos\*\***: Realizar la validación de expertos para asegurar que los requerimientos sean precisos y se ajusten a las necesidades de los usuarios. 4. **\*\*Prototipado\*\***: Crear prototipos de interfaz de usuario para que los stakeholders puedan visualizar y entender mejor los requerimientos. 5. **\*\*Comunicación efectiva\*\***: Fomentar la comunicación efectiva entre los stakeholders, incluyendo a los desarrolladores, usuarios y expertos, para asegurar que todos estén en la misma página. 6. **\*\*Revisión y aprobación\*\***: Realizar una revisión y aprobación cuidadosa de los requerimientos para asegurar que sean precisos y se ajusten a las necesidades de los usuarios. 7. **\*\*Organización jerárquica\*\***: Organizar los requerimientos jerárquicamente, con números y niveles de detalle, para facilitar su gestión y seguimiento. 8. **\*\*Retrospectiva\*\***: Realizar retrospectivas, como en Scrum, para analizar cómo se ha trabajado durante el sprint y identificar oportunidades de mejora. Al implementar estas estrategias, se puede mejorar la comprensión entre los stakeholders y desarrollar soluciones efectivas que se ajusten a las necesidades de los usuarios.

### Respuesta Esperada:

Para evitar la imprecisión en los requerimientos y mejorar la comprensión entre los stakeholders para desarrollar soluciones efectivas, es fundamental asegurarse de que todos los involucrados tengan la misma comprensión de los requerimientos. Esto se puede lograr a través de una comunicación clara y efectiva entre los stakeholders, incluyendo a los desarrolladores, ventas y clientes. Es importante definir los requerimientos de manera precisa y clara, y asegurarse de que todos los involucrados los entiendan de la misma manera. De esta forma, se puede evitar la imprecisión y desarrollar soluciones que se ajusten a las necesidades reales del cliente.

### Contexto Relevante:

1. Imprecisión en los requerimientos Asi fueron definidos los requerimientos Asi lo entendieron los desarrolladores Asi fué resuelto el problema anteriormente Asi fué resuelto el problema ahora Este es e.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....
3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as the first node which mentions avoiding 'imprecision en los requerimientos' by having clear and precise requirements, are ranked higher than the irrelevant nodes, like the second node discussing GET and POST requests and the third node about agile practices in Scrum, which are not related to the input and expected output.

Contextual Recall: 0.75 (Threshold: 0.5) - The score is 0.75 because most sentences in the expected output, such as sentences 1, 2, and 3, are well-supported by node 1 in the retrieval context, which emphasizes clear requirements, but sentence 2 is not well-supported as it discusses communication between stakeholders, which is not mentioned in the context.

Contextual Relevancy: 0.14 (Threshold: 0.5) - The score is 0.14 because the retrieval context is mostly irrelevant to the input, with statements explaining HTTP requests, Scrum practices, and other unrelated topics, as stated in the reasons for irrelevancy, such as 'The statement is not relevant to the input as it explains the use of the GET method in HTTP requests...' and 'The statement is about Scrum retrospective practices, which is not related to avoiding imprecision in requirements and improving stakeholder understanding.' Only a few statements, such as 'imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision' and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores', are relevant to the input.



## Caso de Prueba: 144 - Éxito

### Input del Usuario:

¿Cómo se pueden redactar los requerimientos para evitar la ambigüedad y asegurar que sean comprensibles para usuarios sin conocimientos técnicos?

### Respuesta Generada:

Según el Documento 2, para redactar requerimientos que eviten la ambigüedad y sean comprensibles para usuarios sin conocimientos técnicos, se recomienda: 1. Utilizar un formato estándar y asegurar la adherencia al mismo para todos los requerimientos. 2. Utilizar el lenguaje de manera consistente. 3. Resaltar el texto para distinguir las partes clave del requerimiento. 4. Evitar el uso de jerga informática. Además, se sugiere que un requerimiento se especifique en dos pasos: 1. Realizar una descripción del requerimiento lo más precisa posible. 2. Realizar la especificación de los detalles del requerimiento requeridos al momento de implementar. De esta manera, se puede asegurar que los requerimientos sean claros, concisos y comprensibles para todos los involucrados en el proyecto.

### Respuesta Esperada:

Para redactar los requerimientos y evitar la ambigüedad, es importante que estén escritos de manera clara y concisa, de forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados. Esto se logra utilizando un lenguaje simple y evitando términos técnicos complejos.

### Contexto Relevante:

1. Imprecisión en los requerimientos. Los problemas surgen cuando los requerimientos no son declarados con precisión. Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolla.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing clear and concise information to help users without technical knowledge understand the requirements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are ranked higher than irrelevant nodes, like node 3, which discusses unrelated topics like validation techniques and prototyping, ensuring a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in retrieval context, specifically nodes 1 and 2, which emphasize clear and concise language for non-technical users.

Contextual Relevancy: 0.94 (Threshold: 0.5) - The score is 0.94 because the retrieval context is highly relevant to the input, as it provides specific recommendations for redacting requirements to avoid ambiguity and ensure they are comprehensible to non-technical users, such as 'deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos tecnicos avanzados' and 'usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos'.

## Caso de Prueba: 145 - Éxito

### Input del Usuario:

¿Cuál es la mejor forma de redactar los requisitos para evitar malas interpretaciones y asegurar que sean comprensibles para usuarios sin conocimientos técnicos avanzados?

### Respuesta Generada:

Según el Documento 3, la mejor forma de redactar los requisitos para evitar malas interpretaciones y asegurar que sean comprensibles para usuarios sin conocimientos técnicos avanzados es: 1. Realizar una descripción del requerimiento lo más precisa posible. 2. Realizar la especificación de los detalles del requerimiento requerida al momento de implementar. Además, se recomienda: \* Usar un formato estándar y asegurar la adherencia al mismo para todos los requerimientos. \* Utilizar el lenguaje de manera consistente. \* Resaltar el texto para distinguir las partes clave del requerimiento. \* Evitar el uso de jerga informática. De esta manera, se asegura que los requisitos sean claros, concisos y fácilmente entendibles por todos los involucrados en el proyecto, incluyendo usuarios sin conocimientos técnicos avanzados.

### Respuesta Esperada:

La mejor forma de redactar los requisitos para evitar malas interpretaciones y asegurar que sean comprensibles para usuarios sin conocimientos técnicos avanzados es redactarlos con precisión, evitando la ambigüedad y utilizando un lenguaje claro y sencillo.

### Contexto Relevante:

1. Imprecisión en los requerimientos. Los problemas surgen cuando los requerimientos no son declarados con precisión. Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolla.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. interfaces de usuario la interfaz de usuario ui en una aplicacion es el punto de interaccion visual a traves del cual los usuarios se conectan y utilizan las funcionalidades del software incluye todos.....

3. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer that is easy to understand for users without advanced technical knowledge!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 in the retrieval contexts directly addresses the importance of clear requirements for users without advanced technical knowledge, and node 3 provides specific guidelines for writing clear requirements, but node 2, which is about user interfaces, is ranked higher than it should be, reducing the overall score.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, as the 1st node in the retrieval context explicitly mentions the importance of avoiding ambiguity and using clear language for users without advanced technical knowledge, which is exactly what the output suggests.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context is mostly irrelevant to the input, with statements about user interfaces and design options dominating the context, as seen in reasons such as 'The statement is about user interfaces, which has nothing to do with the input about writing requirements.' However, there are some relevant statements that align with the input, such as 'deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos tecnicos avanzados' and 'recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos', which bring the score up to 0.56.

## Caso de Prueba: 146 - Éxito

### Input del Usuario:

¿Cómo se pueden redactar los requerimientos para evitar la ambigüedad y asegurar que sean comprensibles para todos los involucrados?

### Respuesta Generada:

Según el Documento 2, para redactar los requerimientos de manera clara y evitar la ambigüedad, se recomienda: 1. Utilizar un formato estándar y asegurarse de que todos los requerimientos sigan ese mismo formato. 2. Utilizar un lenguaje consistente y evitar el uso de jerga informática que pueda no ser comprendida por usuarios sin conocimientos técnicos avanzados. 3. Resaltar el texto para distinguir las partes clave del requerimiento. 4. Realizar una descripción del requerimiento lo más precisa posible en dos pasos: \* Primero, se describe el requerimiento de manera general. \* Luego, se proporcionan los detalles específicos del requerimiento que se necesitan para implementarlo. Además, es importante recordar que los requerimientos funcionales se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle (Documento 4). Es fundamental que los requerimientos sean claros y precisos para evitar malentendidos entre los desarrolladores y los usuarios. Al seguir estas recomendaciones, se puede asegurar que los requerimientos sean comprensibles para todos los involucrados y se reduzca la ambigüedad.

### Respuesta Esperada:

Para redactar los requerimientos y evitar la ambigüedad, es importante que estén escritos de manera clara y concisa, de forma que sean comprensibles para todos los involucrados, incluyendo a los usuarios sin conocimientos técnicos avanzados. Esto se logra mediante la utilización de un lenguaje sencillo y evitando términos técnicos complejos. Además, es fundamental ser específico y detallado en la descripción de los requerimientos, para evitar cualquier tipo de interpretación errónea.

### Contexto Relevante:

1. Imprecisión en los requerimientos. Los problemas surgen cuando los requerimientos no son declarados con precisión. Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolla.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 highlighting the importance of precision and node 2 providing specific recommendations for writing requirements, are ranked higher than the irrelevant node 3 discussing non-functional requirements, validation techniques, and prototyping.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in the retrieval context, such as node 1 which emphasizes clear requirements and node 2 which recommends using simple language and being specific, resulting in a flawless match.

Contextual Relevancy: 0.69 (Threshold: 0.5) - The score is 0.69 because the retrieval context is mostly irrelevant, with statements like 'The statement is not directly related to redacting requirements to avoid ambiguity' and 'The statement is about validation of requirements by experts, not about redacting requirements to avoid ambiguity', but some relevant statements like 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios' and 'usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos' were found, which increased the score.

## Caso de Prueba: 147 - Éxito

### Input del Usuario:

¿Cuáles son las consecuencias de tener requerimientos ambiguos en el análisis de requisitos?

### Respuesta Generada:

Según el Documento 1, los requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios. Esto significa que la ambigüedad en los requerimientos puede generar confusión y malentendidos en la comprensión de lo que se desea lograr con el sistema. Las consecuencias de tener requerimientos ambiguos en el análisis de requisitos pueden ser: \* Interpretaciones erróneas de los desarrolladores, lo que puede llevar a una implementación incorrecta del sistema. \* Problemas de comunicación entre los stakeholders (usuarios, desarrolladores, clientes, etc.) debido a la falta de claridad en los requerimientos. \* Dificultades para establecer prioridades y planificar el desarrollo del sistema, ya que no está claro qué se espera lograr. \* Errores en la implementación del sistema, lo que puede llevar a retrasos, sobrecostos y reducir la calidad del producto final. \* Dificultades para realizar pruebas y validación del sistema, ya que no está claro qué se espera que haga el sistema en diferentes situaciones. En resumen, la ambigüedad en los requerimientos puede generar problemas en la comprensión, implementación y validación del sistema, lo que puede llevar a retrasos, sobrecostos y reducir la calidad del producto final. Es fundamental que los requerimientos sean claros, precisos y comprensibles para todos los stakeholders involucrados.

### Respuesta Esperada:

Las consecuencias de tener requerimientos ambiguos en el análisis de requisitos son que pueden interpretarse de diferentes maneras por los desarrolladores y usuarios, lo que puede generar problemas.

### Contexto Relevante:

1. Imprecisión en los requerimientos. Los problemas surgen cuando los requerimientos no son declarados con precisión. Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolla.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.89 (Threshold: 0.5) - The score is 0.89 because the response is mostly relevant, but it could be more direct in addressing the consequences of ambiguous requirements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which directly addresses the question by stating the consequences of having ambiguous requirements, are ranked higher than irrelevant nodes, like node 2 and node 3, which are not related to the topic of consequences of ambiguous requirements in requirement analysis.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly states the consequences of ambiguous requirements.

Contextual Relevancy: 0.89 (Threshold: 0.5) - The score is 0.89 because the retrieval context is mostly relevant, with statements like 'imprecisión en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precisión' and 'requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios' directly addressing the consequences of ambiguous requirements, but it also includes some irrelevant information about non-functional requirements.



## Caso de Prueba: 148 - Éxito

### Input del Usuario:

¿Cómo se puede garantizar que los requerimientos sean lo suficientemente específicos para evitar malentendidos y asegurar que los desarrolladores y usuarios los interpreten correctamente?

### Respuesta Generada:

Según los documentos proporcionados, para garantizar que los requerimientos sean lo suficientemente específicos y evitar malentendidos, se recomienda lo siguiente: 1. **\*\*Redactar los requerimientos de manera clara y concisa\*\***: se deben evitar las palabras ambiguas y se deben utilizar términos técnicos solo cuando sean necesarios, asegurándose de que sean comprensibles para los usuarios sin conocimientos técnicos avanzados (Documento 1). 2. **\*\*Utilizar un formato estándar\*\***: es importante utilizar un formato estándar para todos los requerimientos, asegurándose de que se siga una estructura clara y consistente (Documento 5). 3. **\*\*Especificar los detalles del requerimiento\*\***: cada requerimiento debe ser especificado en dos pasos: primero, se debe realizar una descripción del requerimiento lo más precisa posible, y luego se deben especificar los detalles del requerimiento necesarios para su implementación (Documento 5). 4. **\*\*Incluir archivos y documentación de apoyo\*\***: si es necesario, se deben incluir archivos y documentación de apoyo para proporcionar más contexto y detalles sobre el requerimiento (Documento 2). 5. **\*\*Realizar una validación\*\***: es importante realizar una validación mediante pruebas para asegurarse de que los requerimientos fueron entendidos correctamente y se ajusten a las necesidades del proyecto y del cliente (Documento 3 y 8). 6. **\*\*Identificar y entender las necesidades y expectativas de los usuarios\*\***: es fundamental identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para determinar qué funciones y características debe tener el software (Documento 3, 6 y 8). 7. **\*\*Evitar el uso de jerga informática\*\***: se debe evitar el uso de términos técnicos complejos que puedan confundir a los usuarios, y en su lugar, utilizar un lenguaje claro y fácil de entender (Documento 5). Siguiendo estas recomendaciones, se puede garantizar que los requerimientos sean lo suficientemente específicos para evitar malentendidos y asegurar que los desarrolladores y usuarios los interpreten correctamente.

### Respuesta Esperada:

Para garantizar que los requerimientos sean lo suficientemente específicos, es importante redactarlos de manera clara y concisa, evitando la ambigüedad y asegurando que sean comprensibles para usuarios sin conocimientos técnicos avanzados. De esta forma, se puede evitar la imprecisión y asegurar que los desarrolladores y usuarios los interpreten correctamente.

### Contexto Relevante:

1. Imprecisión en los requerimientos. Los problemas surgen cuando los requerimientos no son declarados con precisión. Requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolla.....

### Contexto Recuperado:

1. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

2. características de la historia de usuario ser lo suficientemente completa como para demostrar el valor para el usuario centrarse en el usuario si no se identifica necesidades precisas se puede iniciar.....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise response without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, directly address the importance of specificity in requirements, highlighting the need for user-centric and clear requirements, and providing a comprehensive overview of the requirement specification process.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which emphasizes the importance of clear and concise requirements for users without advanced technical knowledge, ensuring a precise interpretation by developers and users.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context is perfectly relevant to the input, providing multiple statements that directly address the importance of specific and clear requirements to avoid misunderstandings and ensure correct interpretation by developers and users.

## Caso de Prueba: 149 - Éxito

### Input del Usuario:

¿Cuál es la característica de un requisito que garantiza que solo se incluyan elementos esenciales en el producto?

### Respuesta Generada:

Según el Documento 1, la característica de un requisito que garantiza que solo se incluyan elementos esenciales en el producto es que sea "necesario". Esto significa que un requisito debe ser necesario para el producto correcto, si y solo si cada requisito especificado es un requisito que el software debe hacer. En otras palabras, solo se deben incluir en el producto los elementos que sean verdaderamente necesarios para su funcionamiento correcto.

### Respuesta Esperada:

La característica de un requisito que garantiza que solo se incluyan elementos esenciales en el producto es que sea "Necesario".

### Contexto Relevante:

1. Características de los requerimientos. Necesario: Lo que pida un requisito debe ser necesario para el producto. Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hace.....
2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaracion que identifica una característica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are perfectly relevant to the input, with the first node mentioning 'necesario' as a characteristic, the second node stating 'necesaria para la aceptacion del producto o proceso', and the third node reinforcing this importance, resulting in a flawless ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 2nd node in retrieval context, which emphasizes the necessity of a product or process.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context has some relevant statements, such as 'necesario si y solo si cada requisito especificado es un requisito que el software debe hacer', but is largely irrelevant due to statements like 'el texto debe ser claro preciso y tener una unica interpretacion posible' that are not directly related to the input.

## Caso de Prueba: 150 - Fallo

### Input del Usuario:

¿Cuál es la característica de un requisito que garantiza que solo se incluyan funcionalidades esenciales en el producto?

### Respuesta Generada:

Según el Documento 1, la característica de un requisito que garantiza que solo se incluyan funcionalidades esenciales en el producto es la "necesidad". Es decir, un requisito debe ser necesario para el producto correcto, lo que implica que solo se incluirán funcionalidades esenciales y no se agregará funcionalidades innecesarias.

### Respuesta Esperada:

La característica de un requisito que garantiza que solo se incluyan funcionalidades esenciales en el producto es que sea "Necesario".

### Contexto Relevante:

1. Características de los requerimientos. Necesario: Lo que pida un requisito debe ser necesario para el producto. Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output partially addresses the input question, but includes some irrelevant statements that don't directly describe the characteristic of a requirement.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which mentions 'necesario' as a characteristic of a requirement, and node 2 which aligns with the expected output, are ranked higher than the irrelevant node 3 which does not provide any relevant information.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 2nd node in retrieval context, which explicitly mentions the concept of 'necesaria', directly correlating with the 'Necesario' characteristic in the expected output.

Contextual Relevancy: 0.44 (Threshold: 0.5) - The score is 0.44 because the retrieval context mostly contained irrelevant information, such as 'no ambiguo', 'conciso', 'consistente', 'completo', 'alcanzable', and 'verificable', which are not directly related to the characteristic of a requirement that guarantees only essential functionalities are included in the product, but a few relevant statements like 'necesario si y solo si cada requisito especificado es un requisito que el software debe hacer' and 'caracteristicas de los requerimientos necesario' were present.

## Caso de Prueba: 151 - Éxito

### Input del Usuario:

¿Cuál es la característica fundamental que debe tener un requisito para asegurar que se entienda de manera única y clara?

### Respuesta Generada:

Según el Documento 1, la característica fundamental que debe tener un requisito para asegurar que se entienda de manera única y clara es que sea "no ambiguo". Esto significa que el texto debe ser claro, preciso y tener una única interpretación posible. Esto es esencial para asegurar que todos los stakeholders, incluyendo desarrolladores, clientes y otros interesados, entiendan el requisito de la misma manera y no haya confusiones ni malentendidos.

### Respuesta Esperada:

La característica fundamental que debe tener un requisito para asegurar que se entienda de manera única y clara es que sea "No ambiguo", es decir, que el texto sea claro, preciso y tenga una única interpretación posible.

### Contexto Relevante:

1. Características de los requerimientos. Necesario: Lo que pida un requisito debe ser necesario para el producto. Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 and node 3, which only provide general definitions or duplicate information, are correctly ranked lower than the relevant node 1, which directly answers the input question with a clear characteristic of 'no ambiguo'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly matched with the 1st node in the retrieval context, which explicitly states the same requirement for a clear and unambiguous text.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context is mostly irrelevant, with reasons such as 'the input asks about a fundamental characteristic of a requirement, but this statement talks about...' various aspects of requirements, and only a few statements like 'el texto debe ser claro preciso y tener una unica interpretacion posible' and 'un requisito debe ser no ambigua comprobable o medible y necesaria para la aceptacion del producto o proceso' are relevant to the input.



## Caso de Prueba: 152 - Éxito

### Input del Usuario:

¿Cuál es la característica de un requisito que implica que no debe haber conflicto entre ellos y que el lenguaje empleado sea coherente?

### Respuesta Generada:

Según el contexto proporcionado, la característica de un requisito que implica que no debe haber conflicto entre ellos y que el lenguaje empleado sea coherente es la "consistencia". Esto se menciona en los documentos 1 y 4, que establecen que "ningún requisito debe entrar en conflicto con otro requisito diferente" y que "el lenguaje empleado entre los distintos requisitos debe ser consistente".

### Respuesta Esperada:

La característica de un requisito que implica que no debe haber conflicto entre ellos y que el lenguaje empleado sea coherente es la consistencia.

### Contexto Relevante:

1. Características de los requerimientos. Necesario: Lo que pida un requisito debe ser necesario para el producto. Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the response is mostly relevant, but mentions specific documents, which is not directly relevant to the characteristic of a requirement.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node directly addressing the question and the subsequent nodes being irrelevant due to not mentioning consistency as a characteristic of a requirement.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly matches the 3rd node in retrieval context, which explicitly explains the concept of consistency.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context is only partially relevant, as it provides some related information about requirements, but lacks specific details about the characteristic that implies no conflict and coherent language, as stated in the input.

## Caso de Prueba: 153 - Fallo

### Input del Usuario:

¿Cuál es el propósito de la característica de verificabilidad en los requisitos, y cómo se logra esta verificación?

### Respuesta Generada:

Según el Documento 3, la característica de verificabilidad en los requisitos se refiere a que se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no. El propósito de esta característica es asegurarse de que los requisitos sean cumplidos y que se pueda comprobar que se han alcanzado los objetivos establecidos. La verificación de los requisitos se logra mediante métodos como inspección, análisis, demostración o testeo. Esto permite a los desarrolladores y a los stakeholders evaluar si el sistema cumple con los requisitos establecidos y hacer ajustes necesarios para garantizar que se satisfacen los objetivos del proyecto. En otras palabras, la verificabilidad es una característica fundamental de los requisitos que permite a los desarrolladores evaluar si el sistema cumple con los requisitos y ajustar su desarrollo para asegurarse de que se alcanzan los objetivos establecidos.

### Respuesta Esperada:

El propósito de la característica de verificabilidad en los requisitos es poder determinar con absoluta certeza si un requisito ha sido satisfecho o no. Esta verificación se logra mediante inspección, análisis, demostración o testeo.

### Contexto Relevante:

1. Características de los requerimientos. Necesario: Lo que pida un requisito debe ser necesario para el producto. Correcto: sí y solo sí, cada requisito especificado es un requisito que el software debe.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....
2. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....
3. caracteristicas de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and relevant explanation of the purpose and verification of the verifiability feature in requirements!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, and the irrelevant nodes, such as node 1 with the reason 'requerimientos no funcionales tecnicas...' is not directly related to the purpose of verifiability in requirements, and node 2 with the reason 'clasificacion de los requerimientos...' does not address the purpose of verifiability in requirements, are ranked higher than the relevant node.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which provides a precise match for the given sentences.

Contextual Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the retrieval context contains many irrelevant statements, such as 'recorrido de bpm', which have no connection to the purpose of verifiability in requirements, and only a few relevant statements, like 'verificable se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no', which are related to the input.

## Caso de Prueba: 154 - Fallo

### Input del Usuario:

¿Cuáles son las implicaciones en la experiencia del usuario si el sistema no cumple con los requisitos de usabilidad, como responder rápidamente y recuperarse automáticamente después de un fallo?

### Respuesta Generada:

Si el sistema no cumple con los requisitos de usabilidad, como responder rápidamente y recuperarse automáticamente después de un fallo, las implicaciones en la experiencia del usuario pueden ser significativas. A continuación, se presentan algunas de las posibles consecuencias: 1. **\*\*Frustración y desánimo\*\***: Si el sistema no responde rápidamente, los usuarios pueden sentirse frustrados y desanimados, lo que puede llevar a una disminución en la satisfacción del usuario y la confianza en el sistema. 2. **\*\*Pérdida de productividad\*\***: La falta de respuesta rápida y la recuperación después de un fallo pueden provocar la pérdida de tiempo y productividad, ya que los usuarios tendrán que esperar a que el sistema se recupere o intentar realizar las tareas de nuevo. 3. **\*\*Error humano\*\***: Si el sistema no es intuitivo y no proporciona retroalimentación clara, los usuarios pueden cometer errores al intentar resolver problemas por sí mismos, lo que puede generar más frustración y desánimo. 4. **\*\*Abandono del sistema\*\***: En el peor de los casos, si el sistema no cumple con los requisitos de usabilidad, los usuarios pueden abandonar el sistema y buscar alternativas más fáciles de usar y más confiables. 5. **\*\*Daño a la reputación\*\***: La falta de cumplimiento de los requisitos de usabilidad puede dañar la reputación de la organización que desarrolló el sistema, lo que puede afectar la confianza de los clientes y la lealtad. En resumen, si el sistema no cumple con los requisitos de usabilidad, se puede generar una experiencia del usuario negativa, lo que puede tener consecuencias significativas para la satisfacción del usuario, la productividad y la reputación de la organización.

### Respuesta Esperada:

Si el sistema no cumple con los requisitos de usabilidad, como responder rápidamente y recuperarse automáticamente después de un fallo, las implicaciones en la experiencia del usuario serán negativas. El usuario se sentirá frustrado y desanimado al no obtener respuestas rápidas y al tener que lidiar con fallos que no se recuperan automáticamente. Esto puede llevar a una pérdida de confianza en el sistema y a una experiencia de usuario desagradable.

### Contexto Relevante:

1. Requerimientos no adecuados. El sistema debe ser fácil de utilizar. El sistema proporcionará una respuesta rápida al usuario. El sistema se recuperará automáticamente tras producirse un fallo....

### Contexto Recuperado:

1. tipos de requerimientos no funcionales eficiencia toda funcionalidad del sistema y transaccion de negocio debe responder al usuario en menos de 5 segundos el sistema debe ser capaz de operar adecuadam.....

2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

3. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because the response partially addresses the question, but lacks depth in discussing the implications on user experience and includes some irrelevant statements about different aspects of usability.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which directly addresses usability and node 2 which explains non-functional requirements, are ranked higher than irrelevant nodes, ensuring a perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in retrieval context, specifically node 1, which emphasizes the system's response time and usability, ensuring a seamless user experience.

Contextual Relevancy: 0.24 (Threshold: 0.5) - The score is 0.24 because the retrieval context is mostly irrelevant to the input, with statements like 'seguridad de logica y de datos los permisos de acceso al sistema podran ser cambiados solamente por el administrador del sistema' and 'el sistema debe ser desarrollado usando el lenguaje python 3132' dominating the context, and only a few relevant statements like 'el sistema debe responder al usuario en menos de 5 segundos' and 'el sistema debe ser capaz de operar adecuadamente con hasta 100 usuarios con sesiones concurrentes' that are actually related to user experience implications.

## **Caso de Prueba: 155 - Fallo**

### **Input del Usuario:**

¿Cuál es el requisito clave para que el sistema proporcione una respuesta en tiempo real a los usuarios?

### **Respuesta Generada:**

De acuerdo al Documento 3, el requisito clave para que el sistema proporcione una respuesta en tiempo real a los usuarios es que "su tiempo de respuesta no sera en ningún momento superior a 5 segundos". Esto se considera un requisito no funcional relacionado con la respuesta en el tiempo, lo que se define en el Documento 2 como una propiedad emergente del sistema.

### **Respuesta Esperada:**

El requisito clave para que el sistema proporcione una respuesta en tiempo real a los usuarios es que "el sistema proporcionará una respuesta rápida al usuario".

### **Contexto Relevante:**

1. Requerimientos no adecuados. El sistema debe ser fácil de utilizar. El sistema proporcionará una respuesta rápida al usuario. El sistema se recuperará automáticamente tras producirse un fallo....

### **Contexto Recuperado:**

1. requerimientos no adecuados el sistema debe ser facil de utilizar el sistema proporcionara una respuesta rapida al usuario el sistema se recuperara automaticamente tras producirse un fallo...

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. requerimientos adecuados cuando haya hasta 100 usuarios accediendo simultaneamente al sistema su tiempo de respuesta no sera en ningun momento superior a 5 segundos ante un fallo en el software del si.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addressed the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the most relevant node (rank 1) being a direct match to the expected output, and the irrelevant nodes (ranks 2 and 3) being correctly placed lower due to being related concepts that don't directly address the expected output.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly matches the 1st node in retrieval context, ensuring a rapid response to users.

Contextual Relevancy: 0.27 (Threshold: 0.5) - The score is 0.27 because most of the retrieval context statements were irrelevant to the input, with statements like 'requerimientos no adecuados', 'el sistema debe ser facil de utilizar', and 'reglamentos de seguridad' not related to real-time response, while only a few statements like 'el sistema proporcionara una respuesta rapida al usuario' and 'respuesta en el tiempo' were relevant.



## **Caso de Prueba: 156 - Fallo**

### **Input del Usuario:**

¿Cómo se puede garantizar que el sistema se recupere automáticamente en caso de fallo, si los requisitos no son adecuados?

### **Respuesta Generada:**

Según el Documento 2, los requisitos adecuados establecen condiciones verificables para garantizar la recuperación del sistema en caso de fallo. En este caso específico, se menciona que "ante un fallo en el software del sistema, no se tardará más de 10 minutos en restaurar los datos del sistema en un estado válido y volver a poner en marcha el sistema". Para garantizar que el sistema se recupere automáticamente en caso de fallo, es fundamental establecer requisitos adecuados y verificables, como: 1. Definir un tiempo límite para la restauración del sistema (en este caso, 10 minutos). 2. Establecer un proceso automático para restaurar los datos del sistema en un estado válido. 3. Implementar mecanismos de backup y recuperación de datos para garantizar la integridad de la información. 4. Realizar pruebas y simulaciones para asegurarse de que el sistema se puede recuperar dentro del tiempo límite establecido. En resumen, para garantizar la recuperación automática del sistema en caso de fallo, es crucial definir y establecer requisitos adecuados y verificables, y luego implementar y probar los mecanismos necesarios para cumplir con esos requisitos.

### **Respuesta Esperada:**

Según el contexto, el sistema se recuperará automáticamente tras producirse un fallo, por lo que no es necesario hacer nada adicional para garantizar la recuperación automática, ya que este requisito ya está establecido.

### **Contexto Relevante:**

1. Requerimientos no adecuados. El sistema debe ser fácil de utilizar. El sistema proporcionará una respuesta rápida al usuario. El sistema se recuperará automáticamente tras producirse un fallo....

### **Contexto Recuperado:**

1. requerimientos no adecuados el sistema debe ser facil de utilizar el sistema proporcionara una respuesta rapida al usuario el sistema se recuperara automaticamente tras producirse un fallo...

2. requerimientos adecuados cuando haya hasta 100 usuarios accediendo simultaneamente al sistema su tiempo de respuesta no sera en ningun momento superior a 5 segundos ante un fallo en el software del si.....

3. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant response without any irrelevant statements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly addressing the question and the subsequent nodes being correctly placed lower due to their irrelevance to automatic system recovery in case of failure.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the automatic recovery of the system after a failure.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because the retrieval context mostly discusses non-functional requirements and system usability, with only a few statements mentioning automatic recovery, as seen in 'el sistema se recuperara automaticamente tras producirse un fallo', but lacks a direct connection to the input question.

## Caso de Prueba: 157 - Éxito

### Input del Usuario:

¿Cuál es la estructura óptima para registrar detalles de un requerimiento en un formato que incluya el nombre del requerimiento y su descripción?

### Respuesta Generada:

Según el Documento 1, la estructura óptima para registrar detalles de un requerimiento es la siguiente:

\* **Nombre del requerimiento**: un título breve y descriptivo que identifica el requerimiento. \*

**Descripción**: una descripción detallada del proceso y los detalles del requerimiento que le ayudan al desarrollador a entender con precisión lo que debe programar. Esta estructura se recomienda para registrar los detalles de un requerimiento de manera clara y concisa, facilitando la comprensión del desarrollador y garantizando que se cumplan los requisitos del sistema. Además, según el Documento 2, es importante utilizar un formato estándar y asegurarse de que se utilice un lenguaje consistente para evitar confusiones y asegurar que el requerimiento se especifique de manera precisa.

### Respuesta Esperada:

La estructura óptima para registrar detalles de un requerimiento es la siguiente: Nombre del requerimiento: [Insertar nombre del requerimiento] Descripción: [Insertar descripción del requerimiento] Por ejemplo: Nombre del requerimiento: El sistema debe permitir registrar participantes a un evento Descripción: Proceso de registro de participantes para un evento.

### Contexto Relevante:

1. Formato para registrar detalles de un requerimiento. Nombre del requerimiento El sistema debe permitir registrar participantes a un evento Descripción Proceso • Detalles de los requerimientos que le S.....

### Contexto Recuperado:

1. formato para registrar detalles de un requerimiento nombre del requerimiento el sistema debe permitir registrar participantes a un evento descripcion proceso detalles de los requerimientos que le sirv.....

2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

3. requerimientos en las metodologias agiles historias de usuario hu las historias de usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software una historia debe p.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and relevant response without any irrelevant information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes in the retrieval contexts, such as node 1 with 'formato para registrar detalles de un requerimiento' and node 2 with 'recomendaciones para redactar requerimientos', are correctly ranked higher than the irrelevant node 3 with 'requerimientos en las metodologías ágiles', which is not directly related to the optimal structure for registering requirement details with name and description.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, with sentences 1-2 attributed to node 1, which provides a clear format for registering requirement details.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context mostly contains irrelevant information such as 'el sistema debe permitir registrar participantes a un evento' and 'utilizar el lenguaje de manera consistente', which have no connection to the input about the optimal structure for registering requirement details, but has some relevant statements like 'formato para registrar detalles de un requerimiento' and 'recomendaciones para redactar requerimientos usar un formato estándar'.

## Caso de Prueba: 158 - Éxito

### Input del Usuario:

¿Cuál es el formato óptimo para registrar a los participantes en un evento, considerando los detalles necesarios para una implementación efectiva?

### Respuesta Generada:

Según el Documento 1, el formato óptimo para registrar detalles de un requerimiento es el siguiente: \* Nombre del requerimiento \* El sistema debe permitir [acción específica] \* Descripción \* Proceso \* Detalles de los requerimientos En este caso, para registrar a los participantes en un evento, el formato óptimo sería: \* Nombre del requerimiento: Registrar participantes a un evento \* El sistema debe permitir registrar participantes a un evento \* Descripción: El sistema debe permitir a los auxiliares registrar los participantes a un evento para conocer cuántas personas están inscritas al evento. \* Proceso: El auxiliar ingresa los datos del participante, como nombre, apellido, correo electrónico, teléfono, etc. El sistema valida la información y la registra en una base de datos. \* Detalles de los requerimientos: campos necesarios para registrar a los participantes, como nombre, apellido, correo electrónico, teléfono, fecha de nacimiento, etc. Además, según el Documento 5, es importante resaltar el texto para distinguir las partes clave del requerimiento y evitar el uso de jerga informática. También se recomienda utilizar un lenguaje consistente y realizar una descripción del requerimiento lo más precisa posible. Es importante mencionar que, según el Documento 7, esta historia de usuario se puede descomponer en tareas más específicas, como diseñar e implementar un formulario para capturar los datos del participante, agregar tablas y campos en la base de datos para registrar los datos del formulario, implementar lógica para leer y validar los campos del formulario, etc.

### Respuesta Esperada:

Basado en el contexto proporcionado, el formato óptimo para registrar a los participantes en un evento sería: \*\*Nombre del participante\*\* \*\*Detalles del requerimiento\*\* (en este caso, los detalles necesarios para registrar a los participantes en un evento) Es importante incluir los detalles necesarios para una implementación efectiva, como por ejemplo: \* Nombre y apellido del participante \* Correo electrónico o número de teléfono de contacto \* Fecha de nacimiento (si es relevante para el evento) \* Información adicional específica del evento (por ejemplo, opciones de menú para un evento de comida, etc.) De esta manera, se puede registrar a los participantes de manera clara y concisa, con todos los detalles necesarios para una implementación efectiva del evento.

### Contexto Relevante:

1. Formato para registrar detalles de un requerimiento. Nombre del requerimiento El sistema debe permitir registrar participantes a un evento Descripción Proceso • Detalles de los requerimientos que le s.....

### Contexto Recuperado:

1. formato para registrar detalles de un requerimiento nombre del requerimiento el sistema debe permitir registrar participantes a un evento descripcion proceso detalles de los requerimientos que le sirv.....

2. requerimientos en las metodologías ágiles ejemplos de historias de usuario 1 el sistema debe permitir crear un usuario del sistema tradicional 2 como vendedor deseo registrar los productos y cantidad de.....

3. ejemplos de requerimientos funcionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema debe permitir registrar la venta de.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and comprehensive answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the top 3 nodes directly addressing the question of registering participants in an event, such as the first node being directly related to the question, the second node providing a specific example, and the third node stating a clear requirement.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 3rd node in the retrieval context, which mentions the requirement to register participants in an event.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context has some relevant statements, such as 'el sistema debe permitir registrar participantes a un evento' and 'como auxiliar deseo poder registrar los participantes a un evento para conocer cuantas personas estan inscritas al evento', but is mostly filled with irrelevant information about creating system users, registering products, and canceling debit notes, which are not related to registering participants in an event.

## Caso de Prueba: 159 - Fallo

### Input del Usuario:

¿Qué información mínima se debe registrar para cada participante en un evento?

### Respuesta Generada:

La respuesta a esta pregunta no se encuentra explícitamente en los documentos proporcionados, ya que se centraron en el formato para registrar detalles de un requerimiento, ejemplos de requerimientos funcionales, características de los requerimientos, y otros conceptos relacionados con el desarrollo de software. Sin embargo, podemos inferir que la información mínima que se debe registrar para cada participante en un evento dependerá del propósito del evento y de los requisitos específicos del sistema. En general, se podrían considerar los siguientes campos mínimos: \* Identificador único del participante (por ejemplo, un ID o un número de registro) \* Nombre del participante \* Apellido del participante \* Correo electrónico o dirección de contacto \* Otra información relevante para el evento (por ejemplo, fecha de nacimiento, género, etc.) Es importante destacar que la información mínima necesaria para registrar un participante en un evento puede variar según el contexto y los requisitos del sistema. Es fundamental definir claramente los requisitos funcionales y no funcionales del sistema para determinar qué información es necesaria para registrar a cada participante.

### Respuesta Esperada:

No se proporciona información específica sobre qué información mínima se debe registrar para cada participante en un evento. Sin embargo, se puede inferir que se necesitará registrar al menos el "Nombre del participante" ya que se menciona "Nombre del requerimiento" en el formato para registrar detalles de un requerimiento.

### Contexto Relevante:

1. Formato para registrar detalles de un requerimiento. Nombre del requerimiento El sistema debe permitir registrar participantes a un evento Descripción Proceso • Detalles de los requerimientos que le s.....

### Contexto Recuperado:

1. formato para registrar detalles de un requerimiento nombre del requerimiento el sistema debe permitir registrar participantes a un evento descripcion proceso detalles de los requerimientos que le sirv.....

2. ejemplos de requerimientos funcionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema debe permitir registrar la venta de.....

3. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....

### Métricas de Evaluación:

Answer Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the response contains some relevant information, but is heavily diluted by multiple irrelevant statements that stray away from the question about minimum information required for each participant in an event.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node that implies 'Nombre del participante' is a necessary detail to register for an event, and the second node that is directly related to the question, are ranked higher than the irrelevant node at rank 3, which does not provide any specific information about registering participants in an event.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in the retrieval context, which explicitly mentions 'Nombre del requerimiento' and 'Nombre del participante', making it a seamless match.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because the retrieval context is mostly filled with irrelevant statements about creating system users, registering article sales, canceling debit notes, and contractor payments, which are not related to registering participant information for an event, as stated in the reasons for irrelevancy.



## Caso de Prueba: 160 - Fallo

### Input del Usuario:

¿Cuál es la información mínima necesaria para registrar un participante a un evento?

### Respuesta Generada:

Basándome en la información proporcionada en los documentos, puedo inferir que la información mínima necesaria para registrar un participante a un evento es la siguiente: \* Nombre del participante \* Datos de inscripción al evento (fecha, hora, etc.) \* Información de contacto (dirección, correo electrónico, teléfono, etc.) Esto se desprende del documento 2 y 3, que presentan ejemplos de requerimientos funcionales, incluyendo el registro de participantes a un evento. Además, el documento 8 muestra un ejemplo de descomposición de una historia de usuario en tareas, lo que implica que se necesitan recopilar datos específicos del cliente o participante para registrar su participación en el evento. Sin embargo, es importante destacar que la información necesaria puede variar dependiendo del tipo de evento, la información requerida por el organizador del evento y las necesidades específicas del sistema de registro de participantes.

### Respuesta Esperada:

Según el contexto proporcionado, no se especifica explícitamente qué información mínima es necesaria para registrar un participante a un evento. Sin embargo, se puede inferir que el nombre del participante es un detalle básico y fundamental para registrar a alguien en un evento, por lo que se podría considerar como la información mínima necesaria.

### Contexto Relevante:

1. Formato para registrar detalles de un requerimiento. Nombre del requerimiento El sistema debe permitir registrar participantes a un evento Descripción Proceso • Detalles de los requerimientos que le s.....

### Contexto Recuperado:

1. formato para registrar detalles de un requerimiento nombre del requerimiento el sistema debe permitir registrar participantes a un evento descripcion proceso detalles de los requerimientos que le sirv.....

2. ejemplos de requerimientos funcionales en las metodologías tradicionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema d.....

3. ejemplos de requerimientos funcionales el sistema debe permitir crear un usuario del sistema el sistema debe permitir registrar participantes a un evento el sistema debe permitir registrar la venta de.....

### Métricas de Evaluación:

Answer Relevancy: 0.70 (Threshold: 0.5) - The score is 0.70 because the response is mostly relevant, but includes some tangential information about documents and user stories that aren't directly related to the minimum information needed to register a participant.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node highlighting the need for information to register a participant, the second node emphasizing the importance of registering participants, and the third node reiterating this point, resulting in perfect ranking.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly aligned with the nodes in retrieval context, specifically nodes 2 and 3, which explicitly mention registering participants to an event.

Contextual Relevancy: 0.18 (Threshold: 0.5) - The score is 0.18 because the retrieval context is mostly irrelevant, with many statements not mentioning registering a participant to an event, as stated in the reasons for irrelevancy, and only a few statements, such as 'el sistema debe permitir registrar participantes a un evento', being relevant to the input.

## Caso de Prueba: 161 - Éxito

### Input del Usuario:

¿Cuál es el formato óptimo para registrar detalles de un requerimiento de programación, como el de registrar participantes a un evento, para que el desarrollador pueda entender con precisión lo que se necesita?

### Respuesta Generada:

Según el Documento 1, el formato óptimo para registrar detalles de un requerimiento de programación es: Nombre del requerimiento: [Registrar participantes a un evento] Descripción: [Proceso de registrar participantes a un evento] Detalles de los requerimientos: [Detalles específicos necesarios para implementar el requerimiento, como por ejemplo, campos de entrada, procesos, salida esperada, etc.] De acuerdo con el Documento 2, también es recomendable: \* Utilizar un formato estándar para todos los requerimientos \* Utilizar un lenguaje consistente y claro \* Resaltar el texto para distinguir las partes clave del requerimiento \* Evitar el uso de jerga informática Además, el Documento 4 menciona que los requerimientos funcionales se suelen especificar en lenguaje natural, se expresan de forma individual y se organizan de forma jerárquica a distintos niveles de detalle. También se pueden numerar para facilitar su gestión. En resumen, el formato óptimo para registrar detalles de un requerimiento de programación es aquel que proporciona una descripción clara y precisa del requerimiento, junto con los detalles específicos necesarios para su implementación, utilizando un lenguaje claro y consistente, y organizado de forma jerárquica y estructurada.

### Respuesta Esperada:

Según el contexto, el formato óptimo para registrar detalles de un requerimiento de programación es el siguiente: \* Nombre del requerimiento \* Descripción \* Detalles de los requerimientos Este formato permite al desarrollador entender con precisión lo que se necesita programar.

### Contexto Relevante:

1. Formato para registrar detalles de un requerimiento. Nombre del requerimiento El sistema debe permitir registrar participantes a un evento Descripción Proceso • Detalles de los requerimientos que le S.....

### Contexto Recuperado:

1. formato para registrar detalles de un requerimiento nombre del requerimiento el sistema debe permitir registrar participantes a un evento descripcion proceso detalles de los requerimientos que le sirv.....

2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 which mentions essential components, and node 2 which provides guidelines for writing requirements, are ranked higher than irrelevant nodes, ensuring perfect contextual precision.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 1st node in the retrieval context, which provides a clear and comprehensive format for registering programming requirements.

Contextual Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the retrieval context has some relevant statements, such as 'formato para registrar detalles de un requerimiento' and 'detalles de los requerimientos que le sirven al desarrollador para entender con mas precision lo que hay que programar', but is mostly filled with irrelevant information about coding concepts, software testing, and the software development life cycle, which are not related to registering participants to an event.

## Caso de Prueba: 162 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal del proceso de elicitar requerimientos en el desarrollo de software?

### Respuesta Generada:

De acuerdo con los documentos proporcionados, el propósito principal del proceso de elicitar requerimientos en el desarrollo de software es recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. Este proceso permite identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software, lo que a su vez permite especificar qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. En otras palabras, el propósito principal de la elicitar requerimientos es obtener información precisa y completa sobre los requisitos del sistema de software, lo que garantiza que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente.

### Respuesta Esperada:

El propósito principal del proceso de elicitar requerimientos en el desarrollo de software es recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave.

### Contexto Relevante:

1. Proceso para especificar requerimientos : Sirve para determinar los requerimientos Elicitación: Es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comun.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....

3. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

### Métricas de Evaluación:

Answer Relevancy: 0.86 (Threshold: 0.5) - The score is 0.86 because the response is mostly on point, but slightly deviates from the main purpose of the elicitation process, focusing on its outcome instead.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly addressing the question by defining the main purpose of eliciting requirements, and the irrelevant nodes, such as the second node about course presentation and the third node about analysis, are correctly ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly matched with the 1st node in the retrieval context, which conveys the same information about eliciting software requirements.

Contextual Relevancy: 0.38 (Threshold: 0.5) - The score is 0.38 because most of the retrieval context statements were too broad or unrelated to the purpose of eliciting requirements in software development, as stated in the reasons for irrelevancy, such as 'presentacion de curso', 'caracteristicas del curso', and 'aplicar conceptos metodos y herramientas para crear aplicaciones de software', whereas only a few statements like 'elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software' were relevant to the input.

## **Caso de Prueba: 163 - Fallo**

### **Input del Usuario:**

¿Cuál es el propósito principal del proceso de análisis de requerimientos en el desarrollo de software?

### **Respuesta Generada:**

Según los documentos proporcionados, el propósito principal del proceso de análisis de requerimientos en el desarrollo de software es identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. En específico, el análisis de requerimientos busca determinar los requisitos funcionales y no funcionales del sistema de software, es decir, qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. Esto se logra a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave, y mediante la identificación de actores y funcionalidades. En resumen, el propósito principal del análisis de requerimientos es entender el problema o necesidad de los usuarios y escribir los requerimientos del sistema de software, lo que garantiza que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente.

### **Respuesta Esperada:**

El propósito principal del proceso de análisis de requerimientos en el desarrollo de software es identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software.

### **Contexto Relevante:**

1. Proceso para especificar requerimientos : Sirve para determinar los requerimientos Elicitación: Es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comun.....

### **Contexto Recuperado:**

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....
2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
3. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the output is mostly relevant to the input, but slightly deviates from the main purpose of the requirement analysis process in software development.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which is not related to the main purpose of the requirement analysis process in software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, which exactly matches the sentence.

Contextual Relevancy: 0.41 (Threshold: 0.5) - The score is 0.41 because the retrieval context has many irrelevant statements, such as 'The statement talks about design, which is not directly related to the purpose of requirement analysis.' and 'The statement explains the software development life cycle, which is not directly related to the purpose of requirement analysis.', which dilute the relevance of the few relevant statements.



## Caso de Prueba: 164 - Fallo

### Input del Usuario:

¿Cuál es el propósito del proceso de especificación en el desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, el propósito del proceso de especificación en el desarrollo de software es definir qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. (Documento 6) En otras palabras, la especificación de requerimientos sirve para determinar los requerimientos del sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. Esto implica la identificación de actores y funcionalidades, análisis de requerimientos, especificación y validación mediante pruebas para asegurar que los requerimientos fueron entendidos correctamente. El propósito final de este proceso es garantizar que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente.

### Respuesta Esperada:

El propósito del proceso de especificación en el desarrollo de software es definir qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación, para realizar el documento de especificación de requerimientos.

### Contexto Relevante:

1. Proceso para especificar requerimientos : Sirve para determinar los requerimientos Elicitación: Es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comun.....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....
2. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....
3. conceptos análisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseño es una descripción o especificación de proceso.....

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the output is highly relevant to the input, but slightly loses focus with a tangential mention of interested parties.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'proceso de creacion de software' and 'requerimientos funcionales y no funcionales', is highly relevant, and the third-ranked node, which mentions 'analisis' and 'diseno', is also relevant. However, the second-ranked node, which talks about coding, testing, and the software development life cycle, is an irrelevant node that should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly outlines the software development process and its requirements.

Contextual Relevancy: 0.36 (Threshold: 0.5) - The score is 0.36 because the retrieval context contains mostly irrelevant statements, such as 'presentacion de curso', 'caracteristicas del curso', and 'ideas para recordar de la clase de hoy', which are not related to the purpose of the specification process in software development, and only a few relevant statements like 'proceso de creacion de software' and 'requerimientos funcionales y no funcionales'.

## Caso de Prueba: 165 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal del proceso de validación en el contexto de especificar requerimientos de un sistema de software?

### Respuesta Generada:

Según el contexto proporcionado, el propósito principal del proceso de validación en el contexto de especificar requerimientos de un sistema de software es asegurar que los requerimientos fueron entendidos correctamente, con el fin de garantizar que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente (Documento 1 y Documento 5). En otras palabras, la validación busca verificar que los requerimientos especificados sean correctos y se ajusten a las necesidades y expectativas de los usuarios y partes interesadas, lo que ayudará a garantizar que el sistema de software desarrollado sea el adecuado y cumpla con los objetivos establecidos.

### Respuesta Esperada:

El propósito principal del proceso de validación en el contexto de especificar requerimientos de un sistema de software es asegurar que los requerimientos fueron entendidos correctamente, con el fin de garantizar que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente.

### Contexto Relevante:

1. Proceso para especificar requerimientos : Sirve para determinar los requerimientos Elicitación: Es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comun.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....

3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### Métricas de Evaluación:

Answer Relevancy: 0.86 (Threshold: 0.5) - The score is 0.86 because the response is mostly on point, but slightly deviates from the main purpose of the validation process by mentioning project objectives.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 and node 3, were correctly ranked lower than the relevant node 1, which clearly addresses the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, ensuring a flawless match.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context is perfectly aligned with the input, providing multiple relevant statements that directly address the purpose of the validation process in specifying software system requirements.

## Caso de Prueba: 166 - Éxito

### Input del Usuario:

¿Cuáles son las actividades clave en la etapa de elicitación para obtener los requerimientos del sistema?

### Respuesta Generada:

De acuerdo a los documentos proporcionados, las actividades clave en la etapa de elicitación para obtener los requerimientos del sistema son: 1. Identificación de actores: identificar a los actores clave que están involucrados en el sistema, como usuarios finales, clientes y otros actores clave. 2. Identificación de funcionalidades: determinar qué funcionalidades debe tener el sistema y qué características debe poseer. 3. Entrevistas: realizar diálogos formales o informales con personas para obtener respuestas a un conjunto de preguntas planeadas y entender con claridad cada cosa que el cliente desea. 4. Identificación de personas, sistemas y dispositivos: determinar quiénes o qué sistemas y dispositivos se comunicarán con el sistema y qué se desea que el sistema haga. Estas actividades se mencionan en los documentos 1, 2, 3 y 4.

### Respuesta Esperada:

Las actividades clave en la etapa de elicitación para obtener los requerimientos del sistema son la identificación de actores (personas, otro sistema, dispositivos) y la identificación de funcionalidades (lo que se desea que el sistema haga).

### Contexto Relevante:

1. Proceso para especificar requerimientos Elicitación Actividades Obtención Requerimientos  
Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

3. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is mostly relevant, but slightly derailed by an unnecessary mention of documents, which doesn't directly impact the key

activities in the elicitation phase.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which mentions 'requerimientos no funcionales' and 'entrevista' that are not directly related to the key activities in the elicitation phase.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because although the retrieval context mentioned unrelated terms like 'analisis de requerimientos', 'especificacion', and 'validacion', it also provided relevant information about the elicitation phase, such as 'elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software' and 'identificacion de actores y funcionalidades', which are directly related to the input.

## Caso de Prueba: 167 - Fallo

### Input del Usuario:

¿Cuáles son los tipos de actores que se deben identificar en el proceso de elicitar requerimientos en un proyecto de desarrollo de software?

### Respuesta Generada:

Según el Documento 3 y el Documento 4, en el proceso de elicitar requerimientos en un proyecto de desarrollo de software, se deben identificar los siguientes tipos de actores: 1. Personas: usuarios finales, clientes y otros actores clave. 2. Otro sistema: otros sistemas que interactúan con el sistema de software que se está desarrollando. 3. Dispositivos: dispositivos que interactúan con el sistema de software que se está desarrollando. Es importante identificar a estos actores para entender sus necesidades y expectativas del sistema de software y poder recopilar y descubrir los requerimientos del sistema.

### Respuesta Esperada:

Según el contexto, los tipos de actores que se deben identificar en el proceso de elicitar requerimientos en un proyecto de desarrollo de software son: 1. Personas 2. Otro sistema 3. Dispositivos

### Contexto Relevante:

1. Proceso para especificar requerimientos Elicitación Actividades Obtención Requerimientos  
Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se d.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
2. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....
3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which clearly addresses the question, is followed by an irrelevant node at rank 2,

which is about 'practicas agiles en scrum', unrelated to identifying actors in software development elicitation, and then another relevant node at rank 3, which explicitly lists the types of actors.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 3rd node in the retrieval context, which explicitly lists the required actor types.

Contextual Relevancy: 0.27 (Threshold: 0.5) - The score is 0.27 because the retrieval context mostly contained information about the process of eliciting requirements, specifying requirements, and validation, which are not directly related to the types of actors to identify in the process of eliciting requirements in a software development project, as stated in the input.



## Caso de Prueba: 168 - Éxito

### Input del Usuario:

¿Cuáles son los actores que se deben identificar para especificar los requerimientos en el proceso de elicitar funcionalidades en un sistema?

### Respuesta Generada:

Según los documentos proporcionados, específicamente en los Documentos 2 y 3, para especificar los requerimientos en el proceso de elicitar funcionalidades en un sistema, se deben identificar los siguientes actores: \* Personas (usuarios finales, clientes, etc.) \* Otro sistema \* Dispositivos Estos actores son clave para entender las necesidades y expectativas del sistema y son fundamentales para recopilar y descubrir los requerimientos del sistema de software a través de la comunicación directa. Al identificar correctamente a estos actores, se puede obtener una visión más clara de lo que se desea que el sistema haga y se pueden especificar los requerimientos de manera más precisa.

### Respuesta Esperada:

Los actores que se deben identificar para especificar los requerimientos en el proceso de elicitar funcionalidades en un sistema son: personas, otro sistema, dispositivos y lo que se desea que el sistema haga.

### Contexto Relevante:

1. Proceso para especificar requerimientos Elicitación Actividades Obtención Requerimientos  
Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is highly relevant, but slightly deviates from the input by discussing documents specifying requirements instead of directly identifying actors.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are relevant to the input, with the first node mentioning 'identificacion de actores', the second node highlighting 'identificacion de actores' and 'identificacion funcionalidades', and the third node elaborating on the importance of these aspects and specifying the actors involved.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire sentence in the expected output is perfectly matched with nodes 2 and 3 in the retrieval context, resulting in a flawless alignment.

Contextual Relevancy: 0.65 (Threshold: 0.5) - The score is 0.65 because the retrieval context is only partially relevant, as it contains some relevant information about identifying actors, but also includes irrelevant information about analyzing, specifying, and validating requirements, as stated in the reasons for irrelevancy.

## Caso de Prueba: 169 - Éxito

### Input del Usuario:

¿Cuáles son las actividades clave en el proceso de elicitar requerimientos para identificar lo que se desea que el sistema haga?

### Respuesta Generada:

Según los documentos proporcionados, las actividades clave en el proceso de elicitar requerimientos para identificar lo que se desea que el sistema haga son: 1. Identificación de actores: Identificar a los usuarios finales, clientes y otros actores clave que estarán implicados en el sistema de software. 2. Identificación de funcionalidades: Identificar las funcionalidades y características que se desean en el sistema de software. 3. Entrevistas: Realizar entrevistas con los actores clave para recopilar información sobre los requerimientos del sistema. Esto incluye preparar preguntas que permitan entender con claridad lo que el cliente desea y tomar apuntes o grabar la reunión. 4. Análisis de requerimientos: Analizar las necesidades y expectativas de los usuarios y otras partes interesadas para entender lo que se desea que el sistema haga. 5. Obtención de requerimientos: Obtener información a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. Estas actividades clave se mencionan en varios documentos, incluyendo el Documento 1, Documento 2, Documento 3 y Documento 4.

### Respuesta Esperada:

Las actividades clave en el proceso de elicitar requerimientos para identificar lo que se desea que el sistema haga son la Identificación de actores y la Identificación de Funcionalidades.

### Contexto Relevante:

1. Proceso para especificar requerimientos Elicitación Actividades Obtención Requerimientos  
Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que se d.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

3. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

### Métricas de Evaluación:

Answer Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the output partially addresses the input question, but gets sidetracked by discussing where the activities are mentioned instead of focusing on the activities themselves.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1 to 3, directly address the question by stating the key activities in the elicitation process, making them all highly relevant.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in the retrieval context, specifically the part 'identificacion de actores y funcionalidades', resulting in a flawless match.

Contextual Relevancy: 0.85 (Threshold: 0.5) - The score is 0.85 because the retrieval context contains irrelevant information such as 'proceso para especificar requerimientos' and 'actividades obtencion requerimientos', but still provides relevant statements like 'elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software' and 'lo que se desea que el sistema haga', which align with the input.

## Caso de Prueba: 170 - Fallo

### Input del Usuario:

¿Cuáles son los tipos de actores que se deben identificar durante la obtención de requerimientos en el proceso de elicitar requerimientos?

### Respuesta Generada:

Según el Documento 3 y Documento 5, los tipos de actores que se deben identificar durante la obtención de requerimientos en el proceso de elicitar requerimientos son: \* Personas \* Otro sistema \* Dispositivos Estos actores son importantes para identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas en el sistema de software.

### Respuesta Esperada:

Los tipos de actores que se deben identificar durante la obtención de requerimientos en el proceso de elicitar requerimientos son: \* Personas \* Otro sistema \* Dispositivos

### Contexto Relevante:

1. Proceso para especificar requerimientos Elicitación Actividades Obtención Requerimientos  
Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se d.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
3. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing accurate and relevant information without any irrelevant statements!

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, with the first node ranked low due to being unrelated to identifying actor types, and the second node ranked low as it only discusses user needs without mentioning actor types, whereas the third node, ranked third, correctly identifies 'personas, otro sistema, dispositivos' as actor types.

Contextual Recall: 0.33 (Threshold: 0.5) - The score is 0.33 because the output only partially matches the retrieval context, specifically sentence 1 is attributed to node 3 in the retrieval context, but sentences 2 and 3 lack direct attribution.

Contextual Relevancy: 0.54 (Threshold: 0.5) - The score is 0.54 because the retrieval context contains mostly irrelevant information, such as 'recorrido de bpm', 'estudiar el problema planteado', and 'proceso para especificar requerimientos', which are not related to identifying actor types during requirement elicitation, but has some relevant statements like 'identificacion de actores' and 'personas' that are related to the input.

## Caso de Prueba: 171 - Fallo

### Input del Usuario:

¿Cuál es la técnica de especificación del sistema que implica la creación de una versión preliminar de un sistema para recopilar retroalimentación de los usuarios?

### Respuesta Generada:

Según el contexto proporcionado, la técnica de especificación del sistema que implica la creación de una versión preliminar de un sistema para recopilar retroalimentación de los usuarios es el prototipo. Esto se menciona en el Documento 1, donde se lista el proceso de especificar requerimientos, incluyendo la obtención y validación, y se menciona el uso de prototipos como una de las técnicas para especificar requerimientos. Además, el Documento 2 proporciona más información sobre los prototipos, definiéndolos como "una secuencia de imágenes o prototipos de las interfaces de una app que muestran el conjunto de pantallas o interfaces de usuario de una aplicación" y destacando su papel en la obtención de retroalimentación de los clientes y en la especificación de las funcionalidades de un sistema.

### Respuesta Esperada:

La técnica de especificación del sistema que implica la creación de una versión preliminar de un sistema para recopilar retroalimentación de los usuarios es el Prototipo.

### Contexto Relevante:

1. Proceso para especificar requerimientos Especificación del Sistema Determinar los Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudiar sis.....

### Contexto Recuperado:

1. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....
2. interfaces de usuario interfaces de usuario es una secuencia de imagenes o prototipos de las interfaces de una app que muestran el conjunto de pantallas o interfaces de usuario de una aplicacion las c.....
3. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

### Métricas de Evaluación:

Answer Relevancy: 0.44 (Threshold: 0.5) - The score is 0.44 because the response provides some relevant information about prototypes, but is bogged down by statements that are too broad or tangential to the specific technique asked about.

Contextual Precision: 0.50 (Threshold: 0.5) - The score is 0.50 because the first node, which discusses 'proceso para especificar requerimientos...', is ranked higher than the second node, which clearly describes the correct technique, and the third node, which discusses 'conceptos analisis...', is also ranked higher than the second node, when they should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the information in the 2nd node in retrieval context, which mentions creating a preliminary system to gather user feedback, exactly like the expected output.

Contextual Relevancy: 0.23 (Threshold: 0.5) - The score is 0.23 because the retrieval context is largely irrelevant to the input, with reasons stating 'The input is asking about a specific technique, but this statement is about...' various unrelated topics, and only a few statements, such as 'prototipos' and 'modelo o maqueta del sistema', are remotely relevant.



## Caso de Prueba: 172 - Fallo

### Input del Usuario:

¿Cuál es el propósito de utilizar diagramas visuales como BPM en la especificación del sistema para determinar los requerimientos?

### Respuesta Generada:

Según el Documento 1 y el Documento 5, el propósito de utilizar diagramas visuales como BPM (Business Process Model) en la especificación del sistema para determinar los requerimientos es facilitar la comunicación entre los diferentes roles dentro de una organización y mostrar todo el proceso de una manera clara y comprensible. Esto permite que tanto el personal técnico como no técnico comprendan el flujo del proceso, lo que ayuda a identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. En otras palabras, los diagramas visuales como BPM permiten representar gráficamente los procesos de negocio, lo que facilita la comunicación y el entendimiento de los requerimientos del sistema, lo que a su vez ayuda a determinar los requerimientos del sistema de una manera más efectiva.

### Respuesta Esperada:

El propósito de utilizar diagramas visuales como BPM en la especificación del sistema es para determinar los requerimientos.

### Contexto Relevante:

1. Proceso para especificar requerimientos Especificación del Sistema Determinar los Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudiar sis.....

### Contexto Recuperado:

1. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....

2. requerimientos no adecuados el sistema debe ser facil de utilizar el sistema proporcionara una respuesta rapida al usuario el sistema se recuperara automaticamente tras producirse un fallo...

3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 0.43 (Threshold: 0.5) - The score is 0.43 because the response provided some relevant information, but was often sidetracked by discussing benefits and characteristics of BPM diagrams rather than directly addressing how they help determine system requirements.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval context, which mentions 'diagramas visuales como bpm' and 'determinar los requerimientos', is highly relevant, and the third-ranked node, which mentions 'recorrido de bpm', is also relevant. However, the second-ranked node, which mentions 'requerimientos no adecuados', is not directly related to the purpose of using visual diagrams like BPM in system specification, which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in the retrieval context, ensuring accurate requirement determination.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context mostly contains irrelevant statements that do not specifically mention the purpose of using visual diagrams like BPM, as stated in the reasons for irrelevancy, whereas only a few relevant statements like 'diagramas visuales como bpm' are present.

## Caso de Prueba: 173 - Fallo

### Input del Usuario:

Cuál es la técnica de obtención de información que implica la recopilación de información a través de reuniones grupales y sesiones de brainstorming?

### Respuesta Generada:

La respuesta correcta no se encuentra explícitamente en los documentos proporcionados. Sin embargo, se puede inferir que la técnica de obtención de información que implica la recopilación de información a través de reuniones grupales y sesiones de brainstorming se refiere a la elicitación, mencionada en los Documentos 5 y 6. La elicitación es el proceso de obtención de requerimientos por parte de los actores interesados, como usuarios, clientes o expertos en el dominio. La elicitación se puede realizar a través de varias técnicas, incluyendo reuniones grupales y sesiones de brainstorming, como se menciona en la pregunta. Por lo tanto, aunque no se menciona explícitamente la técnica en los documentos, se puede inferir que la respuesta correcta es la elicitación, que implica la recopilación de información a través de reuniones grupales y sesiones de brainstorming.

### Respuesta Esperada:

La técnica de obtención de información que implica la recopilación de información a través de reuniones grupales y sesiones de brainstorming es el Taller (Lluvia de ideas).

### Contexto Relevante:

1. Proceso para especificar requerimientos Especificación del Sistema Determinar los Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudiar sis.....

### Contexto Recuperado:

1. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....
2. interfaces de usuario interfaces de usuario es una secuencia de imágenes o prototipos de las interfaces de una app que muestran el conjunto de pantallas o interfaces de usuario de una aplicación las c.....
3. estimar historias de usuario prácticas ágiles en scrum la estimación en la metodología ágil consiste en asignar puntos a una tarea o historia de usuario no hay una fórmula para realizarlo de una manera.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the answer is mostly relevant, but slightly derailed by an irrelevant statement that didn't focus on the specific technique of obtaining information asked in the input.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all the top-ranked nodes in the retrieval contexts, including the first node about 'presentacion de curso', the second node about 'interfaces de usuario', and the third node about 'estimar historias de usuario', are irrelevant to the input, failing to mention the technique of obtaining information through group meetings and brainstorming sessions.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the expected output sentence does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.09 (Threshold: 0.5) - The score is 0.09 because the retrieval context is largely irrelevant, with most statements discussing unrelated topics such as user interfaces, software application visuals, and system modeling, as quoted in the reasons for irrelevancy, and only a few statements mentioning techniques for obtaining requirements, which are somewhat related to the input.

## Caso de Prueba: 174 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de una entrevista en el contexto de obtener requerimientos para desarrollar un software?

### Respuesta Generada:

Según el Documento 1, el propósito principal de una entrevista en el contexto de obtener requerimientos para desarrollar un software es identificar el propósito de la entrevista, es decir, buscar respuesta a un conjunto de preguntas planeadas que permitan entender con claridad cada cosa que el cliente desea. En otras palabras, la entrevista busca obtener información precisa y clara sobre los requerimientos del cliente para desarrollar un software que se adapte a sus necesidades.

### Respuesta Esperada:

El propósito principal de una entrevista en el contexto de obtener requerimientos para desarrollar un software es identificar con claridad cada cosa que el cliente desea, es decir, entender sus necesidades y requerimientos.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos La entrevista es un diálogo formal o informal con personas, donde se busca respuesta a un conjunto de preguntas planeadas. • Identifi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
2. objetivos del curso objetivos especificos entender como estimar un producto de software tiempo y valor en de una aplicacion de software desarrollar una aplicacion de software aplicar algunos modelos .....
3. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the actual output partially addresses the input question, but is hindered by circular and incorrect statements that detract from its overall relevance.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 mentioning 'objetivos del curso' and node 3 mentioning 'proceso para especificar

requerimientos', are correctly ranked lower than the relevant node 1, which directly addresses the purpose of an interview in the context of obtaining software requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, capturing the client's needs and requirements with clarity.

Contextual Relevancy: 0.55 (Threshold: 0.5) - The score is 0.55 because many statements in the retrieval context were not directly related to the purpose of an interview in the context of obtaining software requirements, such as 'estudiar el problema planteado', 'familiarizarse con el vocabulario del negocio', and others, which decreased the relevance score.

## Caso de Prueba: 175 - Éxito

### Input del Usuario:

¿Cuáles son los pasos clave para preparar una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales?

### Respuesta Generada:

De acuerdo con el Documento 1, los pasos clave para preparar una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales son: 1. Identificar el propósito de la entrevista: Define claramente lo que se busca lograr en la entrevista y qué tipo de información se necesita recopilar. 2. Preparar preguntas clave: Planea un conjunto de preguntas que permitan entender con claridad cada cosa que el cliente desea y necesita. 3. Identificar posibles entrevistados: Determina quiénes son los funcionarios de diferentes niveles y partes interesadas que pueden brindar información valiosa. 4. Estudiar el problema planteado: Familiarízate con el problema o necesidad que se está tratando de solucionar para que puedas hacer preguntas informáticas y relevantes. 5. Tomar apuntes o grabar la reunión: Asegúrate de registrar la información recopilada durante la entrevista para que puedas referirte a ella más tarde. Es importante recordar que la entrevista es un diálogo formal o informal con clientes y partes interesadas, y su objetivo es recopilar información para crear un software adecuado.

### Respuesta Esperada:

Los pasos clave para preparar una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales son: 1. Identificar el propósito de la entrevista, es decir, definir las preguntas que permitan entender con claridad cada cosa que el cliente desea. 2. Identificar posibles entrevistados, como funcionarios de diferentes niveles. 3. Estudiar el problema planteado para tener una visión clara de lo que se busca resolver. 4. Familiarizarse con el vocabulario del negocio para entender el lenguaje y los conceptos específicos del sector. 5. Prepararse para tomar apuntes o grabar la reunión para registrar la información recopilada durante la entrevista.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos La entrevista es un diálogo formal o informal con personas, donde se busca respuesta a un conjunto de preguntas planeadas. • Identifi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la

comunic.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise response without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts is highly relevant, and the third node is also relevant, but the second node, ranked 2, is an irrelevant node about 'recomendaciones para redactar requerimientos', which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output, including sentences 1-5, can be directly attributed to the 1st node in the retrieval context, demonstrating a perfect alignment between the output and the context.

Contextual Relevancy: 0.55 (Threshold: 0.5) - The score is 0.55 because the retrieval context has some relevant statements about preparing an effective interview with clients and stakeholders, such as 'las entrevistas de los usuarios y las partes interesadas son importantes para crear un software adecuado' and 'preguntas que permitan entender con claridad cada cosa que el cliente desea', but is largely irrelevant due to statements about writing recommendations, using standard formats, and avoiding technical jargon, which are not directly related to the input.



## Caso de Prueba: 176 - Éxito

### Input del Usuario:

¿Cuáles son los pasos clave para prepararse para una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 1, los pasos clave para prepararse para una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales son: 1. Identificar el propósito de la entrevista: Definir claramente qué se busca lograr con la entrevista. 2. Preparar preguntas planeadas: Identificar preguntas que permitan entender con claridad cada cosa que el cliente desea. 3. Identificar posibles entrevistados: Determinar quiénes son las partes interesadas y funcionarios de diferentes niveles que deben ser entrevistados. 4. Estudiar el problema planteado: Familiarizarse con el problema o necesidad que se busca resolver con el desarrollo del software. 5. Familiarizarse con el vocabulario del negocio: Conocer el lenguaje y términos específicos del sector o industria en la que se desenvuelve el cliente. Es importante tener en cuenta que la entrevista es un diálogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas. También es recomendable tomar apuntes o grabar la reunión para asegurarse de capturar toda la información relevante.

### Respuesta Esperada:

Los pasos clave para prepararse para una entrevista efectiva con clientes y partes interesadas para obtener requerimientos no funcionales son: 1. Identificar el propósito de la entrevista: definir preguntas que permitan entender con claridad cada cosa que el cliente desea. 2. Identificar posibles entrevistados: seleccionar funcionarios de diferentes niveles que puedan brindar información valiosa. 3. Estudiar el problema planteado: investigar y comprender el contexto y el problema que se busca resolver. 4. Familiarizarse con el vocabulario del negocio: aprender el lenguaje y los términos específicos del sector o industria para comunicarse efectivamente con los entrevistados. 5. Tomar apuntes o grabar la reunión: registrar la información recopilada durante la entrevista para revisar y analizar posteriormente.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos La entrevista es un diálogo formal o informal con personas, donde se busca respuesta a un conjunto de preguntas planeadas. • Identifi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

2. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

3. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

### **Métricas de Evaluación:**

Answer Relevancy: 0.78 (Threshold: 0.5) - The score is 0.78 because while the output provides some relevant steps to prepare for an effective interview, it includes some irrelevant definitions and characteristics of an interview that detract from its overall relevance.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1 which clearly addresses the question, are ranked higher than the irrelevant nodes, like node 2 and node 3, which only talk about related topics but do not directly provide the steps to prepare for an effective interview.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the 1st node in the retrieval context, demonstrating a flawless contextual recall.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context has some relevant statements, such as 'la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas' and 'preguntas que permitan entender con claridad cada cosa que el cliente desea', but is mostly filled with irrelevant information, like 'The statement mentions 'requerimientos no funcionales tecnicas' which is not directly related to the preparation steps for an effective interview.'

## Caso de Prueba: 177 - Fallo

### Input del Usuario:

¿Cuál es la principal razón por la que se recomienda evitar llevar a cabo entrevistas con los usuarios en grupo al obtener requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 4, no se menciona explícitamente que se recomiende evitar llevar a cabo entrevistas con los usuarios en grupo al obtener requerimientos no funcionales. De hecho, el documento 4 habla sobre la importancia de las entrevistas con los usuarios y las partes interesadas para crear un software adecuado. Sin embargo, podemos inferir que una posible razón por la que se podría evitar entrevistas en grupo es que pueden generar confusiones o contradicciones en las respuestas, lo que podría dificultar la comprensión de los requerimientos no funcionales. Al realizar entrevistas individuales, se puede obtener una mayor claridad y precisión en las respuestas, lo que facilita la comprensión de los requerimientos no funcionales. Es importante destacar que la pregunta no se encuentra directamente respaldada por los documentos proporcionados, por lo que la respuesta se basa en una inferencia lógica.

### Respuesta Esperada:

La principal razón por la que se recomienda evitar llevar a cabo entrevistas con los usuarios en grupo al obtener requerimientos no funcionales es porque no se especifica en el contexto. Sin embargo, se puede inferir que es porque las entrevistas en grupo pueden generar confusión o influir en las respuestas de los usuarios, lo que podría llevar a obtener información no precisa o sesgada.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Entrevistas con los usuarios: Recomendaciones: Evitar...

### Contexto Recuperado:

1. eisc despliegue de aplicaciones web estrategias para actualizar el servidor tipo stand alone stopstart se basa en tener un servidor de produccion sobre el cual se realizan los cambios parar el servido.....
2. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingenieria de software y los elementos metodologicos necesarios para llevar a cabo el desarro.....
3. requerimientos no funcionales tecnicas para obtener requerimientos entrevistas con los usuarios recomendaciones evitar...

### Métricas de Evaluación:

Answer Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because while the output provides some relevant information, it is brought down by the inclusion of irrelevant statements that fail to address the

specific question about avoiding group interviews.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because only one out of three nodes in the retrieval contexts is relevant to the input, with the relevant node being ranked third, and the irrelevant nodes, such as node 1 with the reason 'eisc despliegue de aplicaciones web...' is not related to the topic of avoiding group interviews for non-functional requirements, and node 2 with the reason 'objetivos del curso...' is about software engineering course objectives and does not address the question, are ranked higher.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which clearly discusses 'requerimientos no funcionales' and 'entrevistas con los usuarios', resulting in a flawless match.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context is mostly irrelevant, with statements discussing deployment strategies, software development, and course objectives, as quoted: 'The statement is talking about deployment strategies for web applications, which has no relation to the input about avoiding group interviews for non-functional requirements.' However, a few relevant phrases like 'requerimientos no funcionales', 'tecnicas para obtener requerimientos', and 'entrevistas con los usuarios' were found, but not enough to significantly impact the score.

## Caso de Prueba: 178 - Fallo

### Input del Usuario:

¿Cuál es la principal desventaja de utilizar entrevistas con los usuarios para obtener requerimientos no funcionales?

### Respuesta Generada:

Según el contexto proporcionado, especialmente en el Documento 1, no se menciona explícitamente la principal desventaja de utilizar entrevistas con los usuarios para obtener requerimientos no funcionales. Sin embargo, se puede inferir que una posible desventaja es la imprecisión en los requerimientos, como se menciona en el Documento 6. Los requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrolladores y usuarios, lo que puede llevar a malentendidos y problemas en el proceso de desarrollo del software. Otra posible desventaja no mencionada explícitamente en el contexto es la posibilidad de que los usuarios no estén seguros de lo que necesitan o desean, lo que puede llevar a requerimientos no funcionales vagos o incompletos. Además, las entrevistas pueden ser influenciadas por la forma en que se hacen las preguntas, lo que puede condicionar las respuestas de los usuarios. En resumen, aunque no se menciona explícitamente la principal desventaja, se pueden inferir algunas posibles desventajas, como la imprecisión en los requerimientos y la posible influencia en las respuestas de los usuarios.

### Respuesta Esperada:

La principal desventaja de utilizar entrevistas con los usuarios para obtener requerimientos no funcionales es que no se recomienda.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Entrevistas con los usuarios: Recomendaciones: Evitar...

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
2. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....
3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### Métricas de Evaluación:

Answer Relevancy: 0.70 (Threshold: 0.5) - The score is 0.70 because the output partially addresses the question, but includes some irrelevant statements that don't provide a specific answer to the main disadvantage of using user interviews.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 and node 3, were correctly ranked lower than the relevant node 1, which directly explains the purpose of conducting interviews with users to obtain non-functional requirements, making the ranking perfect.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, resulting in a flawless match!

Contextual Relevancy: 0.08 (Threshold: 0.5) - The score is 0.08 because the retrieval context is mostly irrelevant to the input, with statements like 'identificar el proposito de la entrevista' and 'familiarizarse con el vocabulario del negocio' not addressing the disadvantage of using user interviews to obtain non-functional requirements, and only a few statements like 'entrevistas con los usuarios' being vaguely related.

## Caso de Prueba: 179 - Fallo

### Input del Usuario:

¿Cuáles son algunos temas que debemos evitar discutir durante una entrevista para obtener requerimientos no funcionales con los usuarios?

### Respuesta Generada:

Según el Documento 4, al entrevistar a los usuarios para obtener requerimientos no funcionales, es importante "evitar" ciertos temas. Sin embargo, no se especifican qué temas exactamente deberían evitarse. Sin embargo, podemos inferir que es importante centrarse en la discusión de los requisitos no funcionales, es decir, las limitaciones y restricciones que tienen un impacto en la calidad del sistema, como la seguridad, la escalabilidad, la usabilidad, el rendimiento, entre otros. De esta manera, se pueden evitar discusiones que no aporten valor a la definición de los requisitos no funcionales, como detalles específicos de implementación o características que no son críticas para el sistema. Es importante recordar que la entrevista debe ser un diálogo estructurado, con preguntas planeadas que permitan entender con claridad cada cosa que el cliente desea, y evitar temas que no sean relevantes para la definición de los requisitos no funcionales del sistema.

### Respuesta Esperada:

Algunos temas que debemos evitar discutir durante una entrevista para obtener requerimientos no funcionales con los usuarios son: la tecnología, las soluciones y la implementación.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Entrevistas con los usuarios: Recomendaciones: Evitar...

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....
2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....
3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

### Métricas de Evaluación:

Answer Relevancy: 0.73 (Threshold: 0.5) - The score is 0.73 because the output provides some relevant information, but gets sidetracked with unnecessary details about the interview structure and question planning, which dilutes its focus on the main topic.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 discussing functional requirements and node 3 about agile practices in Scrum, are correctly ranked lower than the relevant node 1, which is directly related to obtaining non-functional requirements with users during an interview.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mentions the formal or informal dialogue with users, directly relating to the discussed topics to avoid during an interview.

Contextual Relevancy: 0.15 (Threshold: 0.5) - The score is 0.15 because the retrieval context contained mostly irrelevant information, such as 'estudiar el problema planteado', 'familiarizarse con el vocabulario del negocio', and 'identificar posibles entrevistados', which have no relation to avoiding topics during an interview for non-functional requirements with users, and only a few relevant statements like 'las entrevistas de los usuarios y las partes interesadas son importantes para crear un software adecuado' and 'preguntas que permitan entender con claridad cada cosa que el cliente desea'.



## Caso de Prueba: 180 - Éxito

### Input del Usuario:

¿Cuáles son las técnicas clave para llevar a cabo una efectiva sesión de lluvia de ideas para obtener requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 1, las técnicas clave para llevar a cabo una efectiva sesión de lluvia de ideas para obtener requerimientos no funcionales son: 1. Tener un moderador que conduzca la sesión y limite el tiempo de discusión para cada parte. 2. Ir preparado con un plan y dividir el problema en partes pequeñas. 3. Hacer diagramas en papel o tableros para aclarar conceptos y capturar la colaboración. 4. Proponer varias alternativas y combinar ideas. 5. Tomar apuntes o grabar la reunión para registrar las ideas y conclusiones. Es importante mencionar que la lluvia de ideas es una técnica utilizada para identificar posibles soluciones a los problemas en aplicaciones nuevas donde hay pocos antecedentes.

### Respuesta Esperada:

Las técnicas clave para llevar a cabo una efectiva sesión de lluvia de ideas para obtener requerimientos no funcionales son: tener un plan y dividir el problema en partes pequeñas, proponer varias alternativas, combinar ideas, tener un moderador y limitar el tiempo de discusión de cada parte, y tomar apuntes o grabar la reunión.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema. • Tener un plan : dividir el problema en partes pequeña.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos Lluvia de ideas reunion donde se proponen ideas para solucionar un problema tener un plan dividir el problema en partes pequenas ej .....

2. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

3. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant, but gets docked for defining brainstorming instead of diving straight into the key techniques for an

effective brainstorming session.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node, which clearly provides the key techniques for conducting an effective brainstorming session to obtain non-functional requirements, ranked highest.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which covers all the key points, resulting in a flawless match.

Contextual Relevancy: 0.65 (Threshold: 0.5) - The score is 0.65 because the retrieval context has many irrelevant statements, such as 'presentacion de curso', 'caracteristicas del curso', and 'diferencias escribir programas vs desarrollar software', which are not related to the input about techniques for non-functional requirements, but has some relevant statements like 'requerimientos no funcionales tecnicas para obtener requerimientos lluvia de ideas' and 'tecnicas para obtener requerimientos no funcionales'.

## Caso de Prueba: 181 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de una reunión de lluvia de ideas en el contexto de obtener requerimientos no funcionales técnicos?

### Respuesta Generada:

De acuerdo al Documento 1, el propósito principal de una reunión de lluvia de ideas en el contexto de obtener requerimientos no funcionales técnicos es identificar posibles soluciones a los problemas para aplicaciones nuevas donde hay pocos antecedentes. La lluvia de ideas se utiliza para proponer varias alternativas y capturar la colaboración entre los participantes, lo que ayuda a aclarar y comprender mejor los problemas y encontrar soluciones efectivas.

### Respuesta Esperada:

El propósito principal de una reunión de lluvia de ideas en el contexto de obtener requerimientos no funcionales técnicos es identificar posibles soluciones a los problemas, especialmente para aplicaciones nuevas donde hay pocos antecedentes.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema. • Tener un plan : dividir el problema en partes pequeña.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos Lluvia de ideas reunion donde se proponen ideas para solucionar un problema tener un plan dividir el problema en partes pequenas ej .....

2. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy dia interactuar con los stake.....

3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### Métricas de Evaluación:

Answer Relevancy: 0.20 (Threshold: 0.5) - The score is 0.20 because the actual output, although related to idea generation, strayed from the main topic of obtaining non-functional technical requirements, focusing on identifying solutions, proposing alternatives, capturing collaboration, and finding effective solutions instead.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the nodes in the retrieval contexts are correctly ranked, with the first node, which directly addresses the question, ranked highest, and the irrelevant nodes, such as the second node discussing the software development life cycle and the third node about specifying requirements, ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the key concepts, resulting in a flawless match.

Contextual Relevancy: 0.34 (Threshold: 0.5) - The score is 0.34 because the retrieval context is largely irrelevant to the input, with most statements being too broad or unrelated to the purpose of a brainstorming meeting in obtaining non-functional technical requirements, as stated in reasons such as 'The statement is too broad and doesn't specifically relate to the purpose of a brainstorming meeting in obtaining non-functional technical requirements.'

## Caso de Prueba: 182 - Éxito

### Input del Usuario:

¿Cuáles son los pasos importantes para asegurar una lluvia de ideas efectiva en una reunión para obtener requerimientos no funcionales técnicos?

### Respuesta Generada:

Según el Documento 1, los pasos importantes para asegurar una lluvia de ideas efectiva en una reunión para obtener requerimientos no funcionales técnicos son: 1. **\*\*Tener un plan\*\***: Dividir el problema en partes pequeñas y tener un moderador que guíe la discusión. 2. **\*\*Ir preparado\*\***: Tener claras las preguntas y objetivos de la reunión. 3. **\*\*Proponer varias alternativas\*\***: Fomentar la creatividad y la colaboración entre los participantes. 4. **\*\*Combinar ideas\*\***: Encontrar soluciones innovadoras y combinar ideas de los participantes. 5. **\*\*Limite el tiempo de discusión\*\***: Establecer un límite de tiempo para cada parte de la discusión para mantener la reunión en marcha. 6. **\*\*Tomar apuntes o grabar la reunión\*\***: Registrar las ideas y conclusiones de la reunión para revisar y refinar posteriormente. Siguiendo estos pasos, se puede asegurar una lluvia de ideas efectiva y productiva en una reunión para obtener requerimientos no funcionales técnicos.

### Respuesta Esperada:

Para asegurar una lluvia de ideas efectiva en una reunión para obtener requerimientos no funcionales técnicos, los pasos importantes son: 1. Tener un plan y dividir el problema en partes pequeñas, utilizando herramientas como diagramas en papel o tableros para aclarar y capturar la colaboración. 2. Proponer varias alternativas y estar preparado para presentarlas. 3. Combinar ideas para generar soluciones innovadoras. 4. Designar un moderador que limite el tiempo de discusión de cada parte y mantenga la reunión enfocada. 5. Tomar apuntes o grabar la reunión para registrar las ideas y conclusiones. Siguiendo estos pasos, se puede garantizar una lluvia de ideas efectiva y obtener requerimientos no funcionales técnicos valiosos.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema. • Tener un plan : dividir el problema en partes pequeña.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos Lluvia de ideas reunion donde se proponen ideas para solucionar un problema tener un plan dividir el problema en partes pequenas ej .....

2. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise response without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as the first node which provides necessary information to answer the question, are ranked higher than the irrelevant nodes, like the second node at rank 2 which is not relevant to brainstorming sessions and the third node at rank 3 which is about Scrum retrospectives.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to the node in the retrieval context, showcasing a seamless alignment between the two.

Contextual Relevancy: 0.55 (Threshold: 0.5) - The score is 0.55 because the retrieval context has some relevant statements, such as 'requerimientos no funcionales tecnicas para obtener requerimientos lluvia de ideas reunion', but is largely irrelevant due to statements like 'The statement is not providing any information about the steps to ensure effective brainstorming in a meeting to obtain non-functional technical requirements.'

## Caso de Prueba: 183 - Éxito

### Input del Usuario:

¿Cuáles son algunas técnicas efectivas para llevar a cabo una lluvia de ideas exitosa en la identificación de soluciones para requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 1, algunas técnicas efectivas para llevar a cabo una lluvia de ideas exitosa en la identificación de soluciones para requerimientos no funcionales son:

1. **\*\*Tener un plan\*\***: Antes de iniciar la lluvia de ideas, es importante tener un plan claro de cómo se va a desarrollar la reunión.
2. **\*\*Dividir el problema en partes pequeñas\*\***: Dividir el problema en partes pequeñas ayuda a aclarar las cosas y a capturar la colaboración de los participantes.
3. **\*\*Hacer diagramas en papel o tableros\*\***: Utilizar diagramas en papel o tableros puede ayudar a aclarar las ideas y a visualizar las soluciones.
4. **\*\*Proponer varias alternativas\*\***: Es importante proponer varias alternativas para que los participantes puedan elegir y desarrollar las mejores ideas.
5. **\*\*Ir preparado\*\***: Es fundamental ir preparado a la reunión con información y conocimientos previos sobre el problema que se va a abordar.
6. **\*\*Combinar ideas\*\***: Combinar las ideas de los participantes puede llevar a soluciones más creativas y efectivas.
7. **\*\*Tener un moderador\*\***: Un moderador puede ayudar a mantener el ritmo de la discusión y a asegurarse de que todas las ideas sean escuchadas.
8. **\*\*Limitar el tiempo de discusión de cada parte\*\***: Es importante limitar el tiempo de discusión de cada parte para mantener el ritmo de la reunión y evitar que se pierda el enfoque. Siguiendo estas técnicas, se puede llevar a cabo una lluvia de ideas exitosa que permita identificar soluciones efectivas para requerimientos no funcionales.

### Respuesta Esperada:

Algunas técnicas efectivas para llevar a cabo una lluvia de ideas exitosa en la identificación de soluciones para requerimientos no funcionales son:

- \* Tener un plan y dividir el problema en partes pequeñas, utilizando herramientas como diagramas en papel o tableros para aclarar y capturar la colaboración.
- \* Proponer varias alternativas y estar preparado para presentarlas.
- \* Combinar ideas para generar soluciones más completas.
- \* Tener un moderador que limite el tiempo de discusión de cada parte y mantenga el enfoque en el objetivo.
- \* Tomar apuntes o grabar la reunión para registrar todas las ideas y no perder detalle alguno. Estas técnicas ayudan a generar un ambiente colaborativo y creativo, donde se pueden identificar posibles soluciones para requerimientos no funcionales.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema. • Tener un plan : dividir el problema en partes pequeña.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos Lluvia de ideas reunion donde se proponen ideas para solucionar un problema tener un plan dividir el problema en partes pequenas ej .....

2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2

prototipa.....

3. objetivos del curso objetivo general proporcionar al estudiante las bases conceptuales fundamentales de la ingeniería de software y los elementos metodológicos necesarios para llevar a cabo el desarrollo.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addressed the input question, providing effective techniques for a successful brainstorming session to identify solutions for non-functional requirements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node that explains techniques for a successful brainstorming session, are ranked higher than irrelevant nodes, like the second and third nodes that discuss unrelated topics like validation techniques and software engineering course objectives.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output, from sentence 1 to 6, are perfectly attributed to the node in the retrieval context, showcasing a seamless alignment.

Contextual Relevancy: 0.74 (Threshold: 0.5) - The score is 0.74 because the retrieval context is mostly irrelevant to the input, with statements like 'course objectives', 'software engineering concepts', 'agile software development methodologies', and 'traditional software development methodologies' having no relation to 'técnicas efectivas para llevar a cabo una lluvia de ideas exitosa', but some relevant statements like 'requerimientos no funcionales técnicas para obtener requerimientos lluvia de ideas' and 'la lluvia de ideas se utiliza para identificar posibles soluciones a los problemas' saved the score from being lower.



## Caso de Prueba: 184 - Éxito

### Input del Usuario:

¿Cuál es la técnica más efectiva para identificar y documentar los requerimientos no funcionales en un modelo de proceso de negocio utilizando BPMN?

### Respuesta Generada:

Según los documentos proporcionados, no hay una técnica específica mencionada como la más efectiva para identificar y documentar los requerimientos no funcionales en un modelo de proceso de negocio utilizando BPMN. Sin embargo, podemos inferir que la técnica de modelos de proceso de negocio utilizando BPMN es efectiva para identificar y documentar los requerimientos funcionales, y por extensión, también puede ser utilizada para identificar y documentar los requerimientos no funcionales. En el Documento 5, se menciona que las historias de usuario se utilizan como una técnica en el proceso de elicitación de requisitos, y que los requisitos se obtienen desde la perspectiva de los usuarios mediante el modelado de procesos de negocio. Esto sugiere que el modelado de procesos de negocio utilizando BPMN puede ser una técnica efectiva para identificar y documentar los requerimientos no funcionales. Además, en el Documento 6, se describe BPMN como un estándar de modelado que proporciona un lenguaje gráfico para documentar procesos de negocio de manera clara y comprensible. Esto sugiere que BPMN puede ser utilizado para modelar procesos de negocio que incluyan requerimientos no funcionales. En cuanto a la documentación de los requerimientos no funcionales, se puede inferir que se puede utilizar una combinación de técnicas, como la elicitación, la identificación de actores y la especificación de requerimientos mencionadas en los documentos 1, 3, 4 y 9. En resumen, aunque no hay una técnica específica mencionada como la más efectiva para identificar y documentar los requerimientos no funcionales en un modelo de proceso de negocio utilizando BPMN, se puede inferir que el modelado de procesos de negocio utilizando BPMN puede ser una técnica efectiva, y que se puede combinar con otras técnicas de elicitación y especificación de requerimientos para identificar y documentar los requerimientos no funcionales.

### Respuesta Esperada:

Según el contexto, la técnica más efectiva para identificar y documentar los requerimientos no funcionales en un modelo de proceso de negocio utilizando BPMN es el BPM (Business Process Model).

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos BPM: Business Process Model. BPMN Business Process Model Notation...

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos la entrevista es un dialogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

2. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the output provides some general information about BPMN and requirements gathering, but lacks specific techniques for identifying and documenting non-functional requirements in a business process model using BPMN.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval context, which mentions 'tecnicas para obtener requerimientos' and 'requerimientos no funcionales', is correctly ranked higher than the second node, which is about a course presentation and software development, and should be ranked lower. However, the second 'no' node is still ranked too high, as it does not provide any direct information about identifying and documenting non-functional requirements in a business process model using BPMN, unlike the third node, which correctly lists 'bpm' as one of the techniques for obtaining requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the sentence perfectly aligns with nodes 1 and 3 in the retrieval context, which mention 'requerimientos no funcionales' and 'BPMN', respectively.

Contextual Relevancy: 0.63 (Threshold: 0.5) - The score is 0.63 because the retrieval context contains many irrelevant statements, such as 'presentacion de curso', 'caracteristicas del curso', and 'desarrollo de software', which are not related to identifying and documenting non-functional requirements in a business process model using BPMN, but also some relevant statements like 'requerimientos funcionales y no funcionales', 'tecnicas para obtener requerimientos', and 'bpm' that make it somewhat relevant.

## Caso de Prueba: 185 - Fallo

### Input del Usuario:

¿Cuál es la técnica más común utilizada para modelar procesos empresariales en BPMN?

### Respuesta Generada:

Según el Documento 3, BPMN (Business Process Model and Notation) es un estándar de modelado desarrollado por la Object Management Group (OMG) que proporciona un lenguaje gráfico para documentar procesos de negocio de manera clara y comprensible. En el Documento 5, se presentan los pasos para extraer historias de usuario desde los procesos de negocio utilizando BPMN. Estos pasos son: 1. Modelar los procesos de negocio utilizando notación BPMN. 2. Organizar el diagrama por carriles. 3. Recorrer el BPM para extraer historias de usuario para cada rol. 4. Realizar un análisis adicional del proceso de negocio en conjunto con usuarios/ clientes para generar nuevas historias de usuario si se requiere modificar el BPM. Por lo tanto, la técnica más común utilizada para modelar procesos empresariales en BPMN es la creación de diagramas de flujo de procesos utilizando la notación BPMN. Esta notación gráfica permite representar los procesos de negocio de manera clara y comprensible, facilitando la comunicación entre los diferentes roles dentro de una organización.

### Respuesta Esperada:

La técnica más común utilizada para modelar procesos empresariales en BPMN es la Notación de Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés).

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos BPM: Business Process Model. BPMN Business Process Model Notation...

### Contexto Recuperado:

1. introduccion a bpmn business process model and notation las historias de usuario desde los procesos de negocio usada como una tecnica en el proceso de elicitation de requisitos los requisitos se obtie.....

2. introduccion a bpmn business process model and notation desafios de esta tecnica decidir el nivel de detalle en que se modelara el proceso de negocio como manejar las variaciones en el proceso de nego.....

3. introduccion a bpm bpm business process model y bpmn business process model es un estandar de modelado desarrollado por la object management group omg que proporciona un lenguaje grafico para document.....

### Métricas de Evaluación:

Answer Relevancy: 0.10 (Threshold: 0.5) - The score is 0.10 because the actual output provides mostly background information and benefits of BPMN, but barely touches on the specific technique used to model business processes in BPMN, making it largely irrelevant to the input question.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the first and third nodes in the retrieval contexts, which mention BPMN and its relation to business process modeling, are correctly ranked higher than the second node, which talks about challenges of a technique and is not directly related to the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which explicitly mentions the key terms 'bpm business process model and notation' and 'modelado de procesos de negocio'.

Contextual Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the retrieval context has some relevant information about BPMN, such as its definition and importance in understanding business processes, but also contains irrelevant information like introductions and user stories, which dilutes its overall relevance to the input.

## Caso de Prueba: 186 - Fallo

### Input del Usuario:

¿Cuál es la relación entre BPMN y los requerimientos no funcionales en un modelo de proceso empresarial?

### Respuesta Generada:

Según los documentos proporcionados, podemos establecer una relación entre BPMN (Business Process Model and Notation) y los requerimientos no funcionales en un modelo de proceso empresarial. BPMN es un estándar de modelado que proporciona un lenguaje gráfico para documentar procesos de negocio de manera clara y comprensible. Esto facilita la comunicación entre los diferentes roles dentro de una organización, permitiendo que tanto el personal técnico como no técnico comprendan el flujo del proceso. Por otro lado, los requerimientos no funcionales se refieren a las restricciones y propiedades emergentes del sistema, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. Estos requerimientos surgen de la necesidad del usuario debido a restricciones en el presupuesto, políticas de la organización, necesidad de interoperabilidad con otros sistemas, entre otros factores internos y externos. La relación entre BPMN y los requerimientos no funcionales se establece a través del proceso de elicitar requisitos. En el documento 10, se menciona que BPMN se utiliza como una técnica en el proceso de elicitar requisitos, obteniendo los requisitos desde la perspectiva de los usuarios mediante el modelado de procesos de negocio. Esto implica que, al modelar el proceso de negocio utilizando BPMN, se pueden identificar los requerimientos no funcionales, como las restricciones y limitaciones del sistema. Además, en el documento 9, se menciona que una de las técnicas para validar requisitos es el recorrido de BPM, que implica hacer un modelo de proceso de negocio que muestre todo el proceso. Esto permite identificar y validar los requerimientos no funcionales, como las limitaciones en el proceso de desarrollo, rendimiento, calidad, normas tecnológicas, etc. En resumen, la relación entre BPMN y los requerimientos no funcionales es que el modelado de procesos de negocio utilizando BPMN facilita la identificación y validación de los requerimientos no funcionales en un modelo de proceso empresarial.

### Respuesta Esperada:

La relación entre BPMN y los requerimientos no funcionales en un modelo de proceso empresarial es que BPMN es una técnica para obtener requerimientos, y dentro de estos requerimientos se encuentran los no funcionales.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos BPM: Business Process Model. BPMN Business Process Model Notation...

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. introducción a bpm bpm business process model y bpmn business process model es un estándar de modelado desarrollado por la object management group omg que proporciona un lenguaje gráfico para

document.....

3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and relevant connection between BPMN and non-functional requirements in a business process model!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node explaining non-functional requirements and the second node introducing BPMN as a standard, are ranked higher than the irrelevant nodes, like the third node which only talks about 'requerimientos funcionales', a related but distinct concept.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions 'requerimientos no funcionales', resulting in a flawless match.

Contextual Relevancy: 0.30 (Threshold: 0.5) - The score is 0.30 because the retrieval context mostly talks about functional requirements, but the input asks about non-functional requirements, as stated in the irrelevant reasons, 'The statement only talks about functional requirements, but the input asks about the relationship between BPMN and non-functional requirements.'

## Caso de Prueba: 187 - Fallo

### Input del Usuario:

¿Cuál es el símbolo utilizado en BPMN para representar una tarea automatizada?

### Respuesta Generada:

De acuerdo al Documento 1, el símbolo utilizado en BPMN para representar una tarea automatizada es la tarea de servicio. Esto se utiliza cuando se invoca un servicio, como un servicio web o una aplicación automática, y no requiere intervención humana. Por ejemplo, como se menciona en el Documento 1, una tarea de script como el envío de correos electrónicos es una tarea automatizada que no requiere interacción humana. Es importante destacar que en BPMN, las tareas se clasifican en diferentes tipos, como tareas manuales, tareas de servicio y tareas de script, cada una con sus características y símbolos representativos.

### Respuesta Esperada:

No se puede determinar el símbolo utilizado en BPMN para representar una tarea automatizada con la información proporcionada en el contexto. El contexto solo menciona que BPMN es la notación para modelar procesos de negocio, pero no proporciona información específica sobre los símbolos utilizados en BPMN.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos BPM: Business Process Model. BPMN Business Process Model Notation...

### Contexto Recuperado:

1. introduccion a bpmn business process model and notation objetos de flujo actividades tareas nombre notacion descripcion invocacion de servicio utiliza algun tipo de servicio este puede ser un servicio.....

2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

3. introduccion a bpmn business process model and notation objetos de flujo actividades tareas nombre notacion descripcion tarea indica un trabajo a realizar es atomica tarea de enviar disenada para envi.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, ranked 1, 2, and 3, are irrelevant to the input, as they do not provide information about the

symbol used in BPMN for an automated task.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the entire expected output does not match any part of the nodes in retrieval context, indicating a complete lack of relevant information.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context is mostly irrelevant to the input, with statements like 'The statement is an introduction to BPMN and does not relate to the symbol for an automated task' and 'The statement lists BPMN objects and does not mention automated tasks or their symbols', but has some relevant information like 'tarea automatizada no tiene intervencion humana' and 'tarea indica un trabajo a realizar es atomica'.



## Caso de Prueba: 188 - Fallo

### Input del Usuario:

¿Cuál es el propósito de la técnica de "Lluvia de ideas)" en el proceso de especificación del sistema?

### Respuesta Generada:

Según el contexto proporcionado, la técnica de "Lluvia de ideas" es utilizada en el proceso de especificación del sistema para recopilar y descubrir los requerimientos del sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. El propósito de esta técnica es generar y recopilar ideas y sugerencias de los stakeholders involucrados en el proyecto, a fin de identificar los requerimientos y necesidades del sistema. En otras palabras, la técnica de "Lluvia de ideas" se utiliza en la etapa de elicitación de requerimientos, con el fin de recopilar información y obtener una visión más completa de lo que se necesita del sistema. Esta técnica permite a los desarrolladores de software obtener una comprensión más clara de las necesidades y expectativas de los usuarios y otras partes interesadas, lo que les permite especificar con mayor precisión los requerimientos del sistema.

### Respuesta Esperada:

El propósito de la técnica de "Lluvia de ideas" en el proceso de especificación del sistema es obtener requerimientos a través de talleres con los usuarios, donde se fomenta la generación de ideas y se recopilan los requisitos del sistema.

### Contexto Relevante:

1. Proceso para especificar requerimientos Especificación del Sistema Determinar los Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudiar sis.....

### Contexto Recuperado:

1. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....
2. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....
3. conceptos la codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

### Métricas de Evaluación:

Answer Relevancy: 0.78 (Threshold: 0.5) - The score is 0.78 because the output is mostly relevant to the input, but slightly derailed by mentioning the technique's usage and types of stakeholders, which aren't

directly addressing the purpose of 'Lluvia de ideas' in system specification.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 mentioning 'presentacion de curso' and node 3 mentioning 'conceptos la codificacion', are correctly ranked lower than the relevant node 1, which explicitly mentions 'proceso para especificar requerimientos' and 'talleres lluvia de ideas', resulting in perfect contextual precision.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions 'talleres' and 'lluvia de ideas', resulting in a flawless match.

Contextual Relevancy: 0.10 (Threshold: 0.5) - The score is 0.10 because the retrieval context is largely irrelevant to the input, with most reasons stating that the statements are too broad, vague, or not related to the purpose of 'Lluvia de ideas' technique, as quoted: 'The statement is too broad and does not specifically relate to the purpose of 'Lluvia de ideas' technique.'

## Caso de Prueba: 189 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de utilizar BPMN en la modelización de procesos empresariales?

### Respuesta Generada:

Según el Documento 1, el propósito principal de utilizar BPMN (Business Process Model and Notation) en la modelización de procesos empresariales es "facilitar la comunicación entre los distintos roles dentro de una organización, permitiendo que tanto el personal técnico como no técnico comprendan el flujo del proceso". En otras palabras, BPMN busca proporcionar un lenguaje gráfico común para documentar procesos de negocio de manera clara y comprensible, permitiendo que todos los involucrados en el proceso empresarial puedan entender y comunicarse efectivamente.

### Respuesta Esperada:

El propósito principal de utilizar BPMN en la modelización de procesos empresariales es representar gráficamente los procesos empresariales de manera estandarizada y clara, facilitando la comprensión y comunicación de los procesos dentro de una organización.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos BPM: Business Process Model. BPMN Business Process Model Notation...

### Contexto Recuperado:

1. introduccion a bpm bpm business process model y bpmn business process model es un estandar de modelado desarrollado por la object management group omg que proporciona un lenguaje grafico para document.....
2. introduccion a bpmn business process model and notation las historias de usuario desde los procesos de negocio usada como una tecnica en el proceso de elicitation de requisitos los requisitos se obtie.....
3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 which clearly states the main goal of using BPMN, are ranked higher than irrelevant nodes, such as node 2 and node 3, which are not directly related to the purpose of using BPMN in business process modeling.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 1st node in the retrieval context, which shares similar ideas of facilitating communication and process comprehension within an organization.

Contextual Relevancy: 0.36 (Threshold: 0.5) - The score is 0.36 because the retrieval context mostly contained irrelevant information, such as 'introduccion a bpm', 'las historias de usuario', and 'agile processes', which are not directly related to the main purpose of using BPMN in business process modeling, and only a few relevant statements, like 'su principal objetivo es facilitar la comunicacion entre los distintos roles dentro de una organizacion', were found.

## Caso de Prueba: 190 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de la validación de expertos en la técnica de validación de requerimientos?

### Respuesta Generada:

Según el Documento 1 y el Documento 7, la validación de expertos es una técnica de validación de requerimientos que consiste en que personas con experiencia revisen los requerimientos y aprueben o rechacen el requerimiento. El propósito principal de la validación de expertos es que los expertos evalúen y aprueben o rechacen los requerimientos recopilados, para asegurar que los requerimientos sean correctos y válidos. En otras palabras, el propósito principal de la validación de expertos es validar que los requerimientos recopilados sean precisos y adecuados, y que reflejen las necesidades y expectativas de los usuarios y otras partes interesadas. De esta manera, se garantiza que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente.

### Respuesta Esperada:

El propósito principal de la validación de expertos en la técnica de validación de requerimientos es que personas con experiencia revisen los requerimientos y aprueben o rechazan el requerimiento.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prototi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....
3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 0.29 (Threshold: 0.5) - The score is 0.29 because the actual output only provides definitions and benefits of expert validation, but fails to address its main purpose, which is the primary focus of the input.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which lacks information about the main purpose of expert validation, and node 3 which only discusses specifying requirements, are correctly ranked lower than the highly relevant node 1 that clearly addresses the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the information in the 1st node in the retrieval context, accurately capturing the expert validation technique.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context has some relevant statements like 'validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento' and 'validacion mediante pruebas para asegurar que los requerimientos fueron entendidos...', but many statements are not directly related to the main purpose of expert validation in requirement validation technique, as they talk about elicitation, specification, requirement analysis, and other unrelated topics.

## Caso de Prueba: 191 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de utilizar prototipos desechables en el prototipado de interfaz de usuario?

### Respuesta Generada:

Según el Documento 1, el propósito principal de utilizar prototipos desechables en el prototipado de interfaz de usuario es para la validación de los requisitos. Estos prototipos se utilizan solo para validar los requisitos y posteriormente se desechan. En otras palabras, se crean para obtener retroalimentación y aprobación de los requisitos y, una vez que se ha alcanzado este objetivo, se descartan.

### Respuesta Esperada:

El propósito principal de utilizar prototipos desechables en el prototipado de interfaz de usuario es la validación de los requisitos.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prototi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. prototipado de interfaz de usuario modelo de pantallas mockup user interfaces fronend ui...
3. modelo de despliegue como calcular el tamano de los servidores de los nodos para despliegue de aplicaciones ram procesadores espacio de almacenamiento ancho de banda herramientas y monitoreo monitoreo.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which directly answers the question, are ranked higher than the irrelevant nodes, such as the second and third nodes which do not provide information about the purpose of using disposable prototypes.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly aligns with the 1st node in the retrieval context, which explicitly mentions the validation of requirements.

Contextual Relevancy: 0.54 (Threshold: 0.5) - The score is 0.54 because the retrieval context has many irrelevant statements, such as 'The statement is about validation of requirements by experts, not about the purpose of disposable prototypes.' and 'The statement talks about deployment models and server sizing, which is not related to the purpose of using disposable prototypes in UI prototyping.', which dilute the relevance of the few relevant statements like 'los dos tipos principales de prototipos de interfaz de usuario son desechables se utilizan solo para la validacion de los requisitos y posteriormente se desechan'.



## Caso de Prueba: 192 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de los prototipos desechables en la validación de requerimientos?

### Respuesta Generada:

Según el Documento 1 y el Documento 9, el propósito principal de los prototipos desechables en la validación de requerimientos es utilizarlos solo para la validación de los requisitos y posteriormente desecharlos. Estos prototipos se utilizan para representar aproximadamente la interfaz de usuario y se pueden crear en papel o en software. Una vez que se han utilizado para la validación de los requisitos, se desechan y no se utilizan más.

### Respuesta Esperada:

El propósito principal de los prototipos desechables en la validación de requerimientos es la validación de los requisitos.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prototi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....
3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which directly answers the question about the main purpose of disposable prototypes in requirement validation, are ranked higher than the irrelevant nodes, which do not mention disposable prototypes or their purpose in requirement validation, like the second and third nodes.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, specifically the 2nd node, which explicitly mentions 'prototipos de interfaz de usuario...son desechables se utilizan solo para la validacion de los requisitos...'.

Contextual Relevancy: 0.14 (Threshold: 0.5) - The score is 0.14 because the retrieval context is mostly irrelevant to the input, with reasons such as 'the statement is too broad and does not specifically relate to the purpose of disposable prototypes in requirement validation' and 'the statement does not mention prototypes or validation', but the few relevant statements like 'prototipos' and 'se utilizan solo para la validacion de los requisitos y posteriormente se desechan' are not enough to make the context highly relevant.

## Caso de Prueba: 193 - Fallo

### Input del Usuario:

¿Cuál es el tipo de prototipo de interfaz de usuario que se mejora su calidad y se convierte progresivamente en el producto final, después de ser utilizado para la validación de los requisitos?

### Respuesta Generada:

Según el Documento 1, el tipo de prototipo de interfaz de usuario que se mejora su calidad y se convierte progresivamente en el producto final, después de ser utilizado para la validación de los requisitos, es el prototipo evolutivo. Los prototipos evolutivos son uno de los dos tipos principales de prototipos de interfaz de usuario, junto con los prototipos desechables. Mientras que los prototipos desechables se utilizan solo para la validación de los requisitos y posteriormente se desechan, los prototipos evolutivos se mejoran su calidad y se convierten progresivamente en el producto final una vez que se han utilizado para la validación de los requisitos.

### Respuesta Esperada:

El tipo de prototipo de interfaz de usuario que se mejora su calidad y se convierte progresivamente en el producto final, después de ser utilizado para la validación de los requisitos, es el Evolutivo.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prototi.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. microservicio microfrontend tipos de estilos arquitecturales para las aplicaciones web microservicio microservicio interfaz de usuario micro componente de la interfaz micro componente de la interfaz m.....
3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because while the output addresses the input question, it includes a tangential statement about disposable prototypes that doesn't directly contribute to the answer.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the first and third nodes in the retrieval contexts, which directly address the question by explaining UI prototypes, are ranked higher than the second node, which is about microservices and microfrontends, an irrelevant topic. However, the score is not higher because the second node, which should be ranked lower, is still present in the top three.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 2nd node in retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.36 (Threshold: 0.5) - The score is 0.36 because the retrieval context is mostly irrelevant, with reasons stating it's about non-functional requirements, expert validation, BPM, and microservices, which are not directly related to the type of UI prototype that improves its quality and becomes the final product, as quoted in the reasons for irrelevancy.

## Caso de Prueba: 194 - Fallo

### Input del Usuario:

¿Cuál es la técnica de validación de requerimientos que implica la creación de un diagrama de flujo que muestra todo el proceso, algoritmo de alto nivel?

### Respuesta Generada:

Según el Documento 3 y Documento 4, la técnica de validación de requerimientos que implica la creación de un diagrama de flujo que muestra todo el proceso, algoritmo de alto nivel se llama "Recorrido de BPM" (Business Process Modeling). Esta técnica se utiliza para validar los requerimientos no funcionales de un sistema de software.

### Respuesta Esperada:

La técnica de validación de requerimientos que implica la creación de un diagrama de flujo que muestra todo el proceso, algoritmo de alto nivel es el Recorrido de BPM.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prototi.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....
2. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....
3. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

### Métricas de Evaluación:

Answer Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the output mentions some related concepts, but gets sidetracked by specific algorithm names and non-functional requirements, diluting its focus on the asked validation technique.

Contextual Precision: 0.33 (Threshold: 0.5) - The score is 0.33 because the top 2 nodes in the retrieval contexts, ranked 1 and 2, are irrelevant nodes, with reasons being 'proceso para especificar requerimientos...' is not directly related to the validation technique mentioned in the expected output and 'recomendaciones para redactar requerimientos...' provides guidelines for writing requirements,

but does not mention the specific validation technique, respectively, and should be ranked lower than the relevant node ranked 3, which mentions 'recorrido de bpm' as a validation technique that involves creating a flowchart showing the entire process, high-level algorithm, which matches the expected output.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output can be accurately attributed to the 3rd node in the retrieval context, perfectly capturing the essence of the Recorrido de BPM technique.

Contextual Relevancy: 0.24 (Threshold: 0.5) - The score is 0.24 because the retrieval context is largely irrelevant to the input, with most statements discussing unrelated topics such as elicitation, identifying actors and functionalities, and writing requirements, as stated in the reasons for irrelevancy, whereas only a few statements, such as '3 recorrido de bpm hacer un bpm que muestre todo el proceso algoritmo de alto nivel resumen', are remotely related to the input about a technique for validating requirements.

## Caso de Prueba: 195 - Fallo

### Input del Usuario:

¿Cuáles son los tipos de requerimientos no funcionales que se consideran críticos en el desarrollo de un sistema de software?

### Respuesta Generada:

Según el contexto proporcionado, específicamente en el Documento 6, se mencionan algunos tipos de requerimientos no funcionales que se consideran críticos en el desarrollo de un sistema de software. Estos son: 1. **\*\*Eficiencia\*\***: La funcionalidad del sistema y las transacciones de negocio deben responder al usuario en menos de 5 segundos, y el sistema debe ser capaz de operar adecuadamente con hasta 100 usuarios con sesiones concurrentes. 2. **\*\*Seguridad\*\***: Los permisos de acceso al sistema solo pueden ser cambiados por el administrador del sistema, y todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA. 3. **\*\*Usabilidad\*\***: El sistema debe contar con manuales de usuario estructurados por cada módulo y funcionalidad, y debe contar con un módulo de ayuda en línea. 4. **\*\*Tecnología\*\***: El sistema debe ser desarrollado usando el lenguaje Python 3.13.2, y debe funcionar en el sistema operativo Linux, Android, Windows y macOS. Es importante destacar que estos requerimientos no funcionales son críticos ya que afectan directamente la experiencia del usuario, la seguridad y la eficiencia del sistema.

### Respuesta Esperada:

Según el contexto, los tipos de requerimientos no funcionales que se consideran críticos en el desarrollo de un sistema de software son: \* No se especifican en el contexto.

### Contexto Relevante:

1. Requerimientos no funcionales Requerimientos no funcionales en las metodologías tradicionales...

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

3. requerimientos no funcionales técnicas para validación de requerimientos 1 validación de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipo.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and comprehensive answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which provides a definition of non-functional requirements and node 2 which explains their purpose, are ranked higher than the irrelevant node 3, which discusses validation techniques, ensuring a perfect ranking.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because none of the sentences in the expected output (1, 2) could be attributed to any node in the retrieval context.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context is only partially relevant, as it discusses non-functional requirements in general, but not specifically critical ones, as stated in the input, and is also mixed with irrelevant information about validation, prototyping, and BPM.



## Caso de Prueba: 196 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de los requerimientos no funcionales en un sistema?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal de los requerimientos no funcionales en un sistema es imponer restricciones y limitaciones en los servicios o funciones ofrecidas por el sistema, relacionadas con propiedades emergentes como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. Estos requerimientos también especifican cómo debe comportarse el sistema en términos de calidad de servicio, rendimiento, seguridad, tecnología, entre otros aspectos. En otras palabras, los requerimientos no funcionales definen las restricciones del sistema, que pueden surgir de la necesidad del usuario, factores internos como el presupuesto y políticas de la organización, y factores externos como reglamentos de seguridad y políticas gubernamentales. En resumen, el propósito principal de los requerimientos no funcionales es establecer las limitaciones y restricciones que el sistema debe cumplir para funcionar correctamente y satisfacer las necesidades del usuario.

### Respuesta Esperada:

El propósito principal de los requerimientos no funcionales en un sistema es especificar "cómo" y "qué tan bien" debe comportarse, imponiendo restricciones que limitan los requerimientos funcionales.

### Contexto Relevante:

1. Requerimientos no funcionales Especifican "qué tan bien" y "como" debe comportarse un sistema Imponen restricciones que típicamente limitan los requerimientos funcionales También conocidos como "requi....."

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestará el sistema en la forma en que reaccionará a determinados insumos generalmente los.....
3. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node explains the purpose of non-functional requirements, but the second-ranked node, which is about functional requirements, should be ranked lower as it's not directly related to the purpose of non-functional requirements, as stated in its reason '...is about functional requirements, which is not directly related to the purpose of non-functional requirements.'

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in retrieval context, which provides a precise match.

Contextual Relevancy: 0.45 (Threshold: 0.5) - The score is 0.45 because the retrieval context has many irrelevant statements that discuss system restrictions, factors leading to non-functional requirements, and functional requirements, as quoted in the reasons for irrelevancy, whereas only a few statements, such as 'requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema...' and 'requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema', are actually relevant to the input.

## Caso de Prueba: 197 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de las restricción en los requerimientos no funcionales de un sistema?

### Respuesta Generada:

Según el Documento 1, los requerimientos no funcionales surgen de la necesidad del usuario debido a restricciones en el presupuesto, políticas de la organización, necesidad de interoperabilidad con otros sistemas de software o hardware, factores externos como reglamentos de seguridad, políticas del gobierno, entre otros. Por lo tanto, el propósito principal de las restricciones en los requerimientos no funcionales de un sistema es limitar o restringir los requerimientos funcionales del sistema, es decir, las funcionalidades específicas que entrega el sistema. Estas restricciones emergen de factores internos y externos y tienen como objetivo imponer límites o condicionantes en la diseño y desarrollo del sistema, con el fin de garantizar que el sistema cumpla con ciertos atributos de calidad, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento, entre otros. En otras palabras, el propósito principal de las restricciones en los requerimientos no funcionales es asegurar que el sistema se comporte de una manera específica y cumpla con ciertos estándares de calidad, seguridad y performance, lo que a su vez impacta en la forma en que se diseñan y desarrollan las funcionalidades del sistema.

### Respuesta Esperada:

El propósito principal de las restricciones en los requerimientos no funcionales de un sistema es imponer límites a los requerimientos funcionales.

### Contexto Relevante:

1. Requerimientos no funcionales Especifican "qué tan bien" y "como" debe comportarse un sistema Imponen restricciones que típicamente limitan los requerimientos funcionales También conocidos como "requi....."

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....
3. requerimientos no funcionales técnicas para obtener requerimientos la entrevista es un diálogo formal o informal con personas donde se busca respuesta a un conjunto de preguntas planeadas identificar .....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the response partially addresses the input question, but gets sidetracked by discussing the origin of non-functional requirements, which is not the main focus of the question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like node 1 and node 2, are ranked higher than the irrelevant node 3, which is about obtaining requirements through interviews and does not address the purpose of non-functional requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, as the 1st node in the retrieval context directly mentions key terms like 'requerimientos no funcionales' and 'restricciones', making it an ideal match.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context mostly consists of irrelevant statements, such as 'The statement is about the capacity of input/output devices and data representation in the system interface...' and 'The statement is talking about identifying possible interviewees...', which are not related to the main purpose of non-functional requirements, making the relevant statements, like 'requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas...', get lost in the noise.

## Caso de Prueba: 198 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de los atributos de calidad en los requerimientos no funcionales de un sistema?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal de los atributos de calidad en los requerimientos no funcionales de un sistema es establecer restricciones y limitaciones en el sistema, y definir cómo debe comportarse y responder en diferentes situaciones. En específico, los atributos de calidad se refieren a características como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, la escalabilidad, la seguridad, la mantenibilidad, la interoperabilidad, entre otras. Estos atributos emergentes del sistema determinan cómo el sistema debe funcionar y responder en diferentes condiciones, y están influenciados por factores internos y externos, como la necesidad del usuario, la política de la organización, la interoperabilidad con otros sistemas, y los reglamentos de seguridad, entre otros. En resumen, el propósito principal de los atributos de calidad en los requerimientos no funcionales es definir y establecer las restricciones y limitaciones del sistema, y garantizar que se cumplan ciertos estándares de calidad y rendimiento.

### Respuesta Esperada:

El propósito principal de los atributos de calidad en los requerimientos no funcionales de un sistema es especificar "cómo" y "qué tan bien" debe comportarse un sistema.

### Contexto Relevante:

1. Requerimientos no funcionales Especifican "qué tan bien" y "como" debe comportarse un sistema Imponen restricciones que típicamente limitan los requerimientos funcionales También conocidos como "requi....."

### Contexto Recuperado:

1. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....
2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....
3. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....

### Métricas de Evaluación:

Answer Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the output provides some relevant information about quality attributes, but is heavily diluted by multiple irrelevant statements about emergent attributes and their influencing factors.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like the first node which directly answers the question, are ranked higher than irrelevant nodes, like the second and third nodes which do not explicitly state the purpose of quality attributes.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, which clearly states the purpose of non-functional requirements.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context is mostly irrelevant, with statements like 'factores internos', 'factores externos', 'clasificacion de los requerimientos', and 'los requerimientos funcionales...' not related to the main purpose of quality attributes in non-functional requirements, but some relevant statements like 'requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema' and 'requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema' were present.

## Caso de Prueba: 199 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de los requisitos técnicos en el diseño de un sistema?

### Respuesta Generada:

Según el Contexto, el propósito principal de los requisitos no funcionales, también conocidos como requisitos técnicos, es imponer restricciones que limitan los requerimientos funcionales del sistema. Estos requisitos especifican cómo debe comportarse el sistema en términos de tiempo, rendimiento, calidad, normas tecnológicas, entre otros aspectos. En otras palabras, los requisitos técnicos definen los atributos de calidad o requisitos de calidad de servicio que deben cumplir los sistemas, lo que permite evaluar el desempeño del sistema en términos de eficiencia, efectividad y satisfacción del usuario. En resumen, el propósito principal de los requisitos técnicos es establecer las limitaciones y restricciones que garantizan que el sistema se comporte de manera efectiva y eficiente, cumpliendo con los estándares y expectativas establecidos.

### Respuesta Esperada:

El propósito principal de los requisitos técnicos en el diseño de un sistema es especificar "qué tan bien" y "como" debe comportarse un sistema, imponiendo restricciones que limitan los requerimientos funcionales.

### Contexto Relevante:

1. Requerimientos no funcionales Especifican "qué tan bien" y "como" debe comportarse un sistema Imponen restricciones que típicamente limitan los requerimientos funcionales También conocidos como "requi....."

### Contexto Recuperado:

1. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....
2. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....
3. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant, but slightly derailed by mentioning non-functional requirements, which aren't the focus of the question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which clearly explains the purpose of technical requirements, are ranked higher than the irrelevant nodes, such as the second and third nodes which talk about unrelated topics like agile development and requirement classification.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the expected output, as seen in the 1st node which explicitly states the purpose of technical requirements in system design, mirroring the sentence.

Contextual Relevancy: 0.71 (Threshold: 0.5) - The score is 0.71 because although the retrieval context does not directly mention the purpose of technical requirements in system design, it does contain relevant statements such as 'tambien conocidos como requisitos tecnicos' and 'requerimientos no funcionales' that are related to the input, indicating some connection between the two.



## Caso de Prueba: 200 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de los requisitos de calidad de servicio en un sistema?

### Respuesta Generada:

Según los documentos proporcionados, los requisitos de calidad de servicio, también conocidos como requisitos no funcionales, tienen como propósito principal imponer restricciones y limitaciones en los servicios o funciones ofrecidas por el sistema. Estos requisitos establecen atributos de calidad como tiempos de respuesta, rendimiento, calidad, normas tecnológicas, entre otros, que deben ser cumplidos por el sistema durante su funcionamiento. En otras palabras, el propósito principal de los requisitos de calidad de servicio es definir cómo debe comportarse un sistema en términos de performance, escalabilidad, seguridad, mantenimiento, entre otros aspectos, para garantizar que el sistema cumpla con las expectativas de los usuarios y se ajuste a las necesidades del negocio. Por ejemplo, según el Documento 7, un requisito no funcional es que el sistema tenga un tiempo de respuesta no superior a 5 segundos cuando haya hasta 100 usuarios accediendo simultáneamente al sistema. Este requisito de calidad de servicio establece una limitación en el rendimiento del sistema que debe ser cumplido durante su funcionamiento.

### Respuesta Esperada:

El propósito principal de los requisitos de calidad de servicio en un sistema es especificar "qué tan bien" y "como" debe comportarse un sistema.

### Contexto Relevante:

1. Requerimientos no funcionales Especifican "qué tan bien" y "como" debe comportarse un sistema Imponen restricciones que típicamente limitan los requerimientos funcionales También conocidos como "requi....."

### Contexto Recuperado:

1. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que tipicamente limitan los requerimientos funcionales tambien conocidos como requisitos.....
2. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....
3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the actual output contains several irrelevant statements that fail to address the main purpose of quality of service requirements, despite providing some related information.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly answering the question about the purpose of quality of service requirements, and the lower-ranked nodes 2 and 3 being correctly placed below as they are not relevant to the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly aligns with the 1st node in the retrieval context, which explicitly states the main purpose of quality service requirements.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context is only partially relevant to the input, as it discusses non-functional requirements, which are related to quality of service, but the majority of the statements are focused on classification and functional requirements, as stated in the reasons for irrelevancy, such as 'The input asks about the main purpose of quality of service requirements, but the statement only talks about classification of requirements.'

## **Caso de Prueba: 201 - Fallo**

### **Input del Usuario:**

¿Cuál es el límite de respuesta para cada funcionalidad del sistema y transacción de negocio en términos de segundos?

### **Respuesta Generada:**

De acuerdo al Documento 1, el límite de respuesta para cada funcionalidad del sistema y transacción de negocio es de menos de 5 segundos. Esto significa que el sistema debe ser capaz de responder a las solicitudes de los usuarios en un tiempo máximo de 5 segundos.

### **Respuesta Esperada:**

El límite de respuesta para cada funcionalidad del sistema y transacción de negocio es de menos de 5 segundos.

### **Contexto Relevante:**

1. Tipos de requerimientos no funcionales Eficiencia: Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos. El sistema debe ser capaz de operar adecu.....

### **Contexto Recuperado:**

1. tipos de requerimientos no funcionales eficiencia toda funcionalidad del sistema y transaccion de negocio debe responder al usuario en menos de 5 segundos el sistema debe ser capaz de operar adecuadam.....

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. requerimientos adecuados cuando haya hasta 100 usuarios accediendo simultaneamente al sistema su tiempo de respuesta no sera en ningun momento superior a 5 segundos ante un fallo en el software del si.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts directly answers the question, and the third-ranked node also provides a clear response time, but the second-ranked node is an irrelevant node that discusses non-functional requirements, which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in the retrieval context, which explicitly states the response time requirement.

Contextual Relevancy: 0.19 (Threshold: 0.5) - The score is 0.19 because the retrieval context is mostly irrelevant to the input, with reasons such as 'the input is asking about response time limit, but the statement is about [various unrelated topics]' and 'the retrieval context contained information about [unrelated topics] which has nothing to do with response time in seconds', with only a few relevant statements that mention response time, like 'toda funcionalidad del sistema y transaccion de negocio debe responder al usuario en menos de 5 segundos'.

## Caso de Prueba: 202 - Fallo

### Input del Usuario:

¿Cuál es el algoritmo de encriptación que se utilizará para las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema?

### Respuesta Generada:

Según el Documento 4, se menciona que "todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA". Por lo tanto, el algoritmo de encriptación que se utilizará para las comunicaciones externas es el algoritmo RSA.

### Respuesta Esperada:

El algoritmo de encriptación que se utilizará para las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema es RSA.

### Contexto Relevante:

1. Tipos de requerimientos no funcionales Eficiencia: Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos. El sistema debe ser capaz de operar adecu.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

3. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer without any irrelevant information.

Contextual Precision: 0.00 (Threshold: 0.5) - The score is 0.00 because all nodes in the retrieval contexts, including the 1st node discussing non-functional requirements, the 2nd node being a duplicate, and the 3rd node explaining HTTP requests, are irrelevant to the input and should be ranked lower.

Contextual Recall: 0.00 (Threshold: 0.5) - The score is 0.00 because the expected output sentence does not match any part of the nodes in retrieval context.

Contextual Relevancy: 0.00 (Threshold: 0.5) - The score is 0.00 because the retrieval context does not mention encryption algorithms and is filled with irrelevant information such as 'requerimientos no funcionales tecnicas para validacion de requerimientos', 'validacion de expertos', and 'prototipado de interfaz de usuario', which are not related to the input's question about encryption algorithm.

## Caso de Prueba: 203 - Fallo

### Input del Usuario:

¿Cuál es el tiempo máximo de respuesta del sistema para cada funcionalidad y transacción de negocio según el requerimiento de eficiencia?

### Respuesta Generada:

Según el Documento 3, el requerimiento de eficiencia establece que "toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos". Por lo tanto, el tiempo máximo de respuesta del sistema para cada funcionalidad y transacción de negocio es de 5 segundos.

### Respuesta Esperada:

Según el requerimiento de eficiencia, el tiempo máximo de respuesta del sistema para cada funcionalidad y transacción de negocio es de menos de 5 segundos.

### Contexto Relevante:

1. Tipos de requerimientos no funcionales Eficiencia: Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos. El sistema debe ser capaz de operar adecu.....

### Contexto Recuperado:

1. requerimientos adecuados cuando haya hasta 100 usuarios accediendo simultaneamente al sistema su tiempo de respuesta no sera en ningun momento superior a 5 segundos ante un fallo en el software del si.....

2. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones especificas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

3. tipos de requerimientos no funcionales eficiencia toda funcionalidad del sistema y transaccion de negocio debe responder al usuario en menos de 5 segundos el sistema debe ser capaz de operar adecuadam.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 directly answers the question, and node 3 also provides the specific answer, but node 2, which explains a concept, is ranked second and should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the sentence perfectly matches the 3rd node in the retrieval context, ensuring a flawless recall.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context is mostly irrelevant, with statements like 'la capacidad de almacenamiento de forma alternativa definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida' and 'usabilidad el sistema debe contar con manuales de usuario estructurados por cada modulo y funcionalidad' that have no relation to the input, but contains some relevant statements like 'requerimientos adecuados cuando haya hasta 100 usuarios accediendo simultaneamente al sistema su tiempo de respuesta no sera en ningun momento superior a 5 segundos'.



## Caso de Prueba: 204 - Fallo

### Input del Usuario:

¿Cuál es el lenguaje de programación que se debe utilizar para desarrollar el sistema según los requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 1, los requerimientos no funcionales establecen que "el sistema debe ser desarrollado usando el lenguaje python 3.13.2". Por lo tanto, el lenguaje de programación que se debe utilizar para desarrollar el sistema es Python 3.13.2.

### Respuesta Esperada:

Según los requerimientos no funcionales, el lenguaje de programación que se debe utilizar para desarrollar el sistema es Python 3.13.2.

### Contexto Relevante:

1. Tipos de requerimientos no funcionales Eficiencia: Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos. El sistema debe ser capaz de operar adecu.....

### Contexto Recuperado:

1. tipos de requerimientos no funcionales eficiencia toda funcionalidad del sistema y transaccion de negocio debe responder al usuario en menos de 5 segundos el sistema debe ser capaz de operar adecuadam.....

2. requerimientos no adecuados el sistema debe ser facil de utilizar el sistema proporcionara una respuesta rapida al usuario el sistema se recuperara automaticamente tras producirse un fallo...

3. recomendaciones para redactar requerimientos usar un formato estandar y asegurar la adherencia al mismo para todos los requerimientos utilizar el lenguaje de manera consistente resaltar el texto para .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, like the first node which clearly addresses the question, are ranked higher than the irrelevant nodes, such as the second node which doesn't mention the programming language to be used, and the third node which is about writing requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the required

technology and Python version.

Contextual Relevancy: 0.05 (Threshold: 0.5) - The score is 0.05 because the retrieval context mostly contains irrelevant statements that don't mention the programming language required for system development, as stated in the reasons for irrelevancy, such as 'The statement does not mention the programming language required for system development.' and 'This statement talks about the system's ability to handle concurrent users, not the programming language.', with only one relevant statement 'el sistema debe ser desarrollado usando el lenguaje python'.

## Caso de Prueba: 205 - Éxito

### Input del Usuario:

¿Cuál es el factor no funcional que se refiere a la capacidad del sistema para funcionar correctamente y sin errores?

### Respuesta Generada:

Según el contexto proporcionado, el factor no funcional que se refiere a la capacidad del sistema para funcionar correctamente y sin errores es la "fiabilidad". La fiabilidad se menciona en el Documento 1 como una de las propiedades emergentes de un sistema, que se refiere a la capacidad del sistema para funcionar de manera confiable y sin errores.

### Respuesta Esperada:

Según el contexto, el factor no funcional que se refiere a la capacidad del sistema para funcionar correctamente y sin errores es la "fiabilidad".

### Contexto Relevante:

1. Requerimientos no funcionales Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

3. clasificacion de los requerimientos los requerimientos funcionales declaraciones de los servicios que debe proporcionar el sistema la forma en que el sistema debe reaccionar a las entradas y la forma .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and accurate answer without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as the 2nd node mentioning 'requerimientos funcionales...' and the 3rd node about 'clasificacion de los requerimientos...', are correctly ranked lower than the 1st node that directly addresses the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, which explicitly mentions 'la fiabilidad' as a non-functional requirement, resulting in a flawless contextual recall.

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context mostly talks about classification of requirements, functional requirements, and their characteristics, whereas the input specifically asks about a non-functional factor, making most of the context irrelevant, as stated in the reasons for irrelevancy, such as 'The input is asking about a specific non-functional factor, but this statement is about classification of requirements in general.' and 'The input is asking about non-functional requirements, but this statement is about functional requirements.'

## Caso de Prueba: 206 - Fallo

### Input del Usuario:

¿Cuál es el requerimiento no funcional que se refiere a la velocidad a la que un sistema responde a las solicitudes de los usuarios?

### Respuesta Generada:

Según los documentos proporcionados, específicamente en el Documento 1, se menciona que los requerimientos no funcionales se refieren a las propiedades emergentes del sistema, como la respuesta en el tiempo. Por lo tanto, el requerimiento no funcional que se refiere a la velocidad a la que un sistema responde a las solicitudes de los usuarios es la "respuesta en el tiempo". Además, en el Documento 10, se proporciona un ejemplo de requerimiento no funcional que se refiere específicamente a la respuesta en el tiempo, estableciendo que el tiempo de respuesta no será superior a 5 segundos cuando haya hasta 100 usuarios accediendo simultáneamente al sistema.

### Respuesta Esperada:

El requerimiento no funcional que se refiere a la velocidad a la que un sistema responde a las solicitudes de los usuarios es la "respuesta en el tiempo".

### Contexto Relevante:

1. Requerimientos no funcionales Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. requerimientos no funcionales especifican que tan bien y como debe comportarse un sistema imponen restricciones que típicamente limitan los requerimientos funcionales también conocidos como requisitos.....

3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the response is mostly relevant, but could be more specific in addressing the input about response speed.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node mentioning 'respuesta en el tiempo', are ranked higher than irrelevant nodes, like the second

node and third node, which do not directly address the speed of the system's response to user requests.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the information provided in the 1st node in retrieval context, which explicitly mentions 'respuesta en el tiempo'.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because the retrieval context only mentions 'respuesta en el tiempo' which is related to 'velocidad a la que un sistema responde a las solicitudes de los usuarios', but the context lacks direct relevance to the input.

## Caso de Prueba: 207 - Fallo

### Input del Usuario:

¿Cuál es el impacto de la capacidad de almacenamiento en la respuesta en el tiempo y su relación con los requerimientos no funcionales?

### Respuesta Generada:

Según los documentos proporcionados, el impacto de la capacidad de almacenamiento en la respuesta en el tiempo se relaciona directamente con los requerimientos no funcionales. En específico, el Documento 2 menciona que la capacidad de almacenamiento es un requisito no funcional que debe ser evaluado para determinar cuánto espacio en disco se necesita no solo para la instalación de la aplicación sino también para el almacenamiento de datos, bases de datos, logs y archivos generados por usuarios, etc. Esto implica que la capacidad de almacenamiento puede afectar la respuesta en el tiempo de la aplicación, ya que si no hay suficiente espacio de almacenamiento, la aplicación puede ralentizarse o incluso fallar. Además, el Documento 7 proporciona un ejemplo de requerimiento no funcional relacionado con la respuesta en el tiempo, que establece que el tiempo de respuesta no debe ser superior a 5 segundos cuando hasta 100 usuarios acceden simultáneamente al sistema. Esto sugiere que la capacidad de almacenamiento es un factor importante para garantizar que la aplicación responda rápidamente y eficientemente. En resumen, la capacidad de almacenamiento tiene un impacto significativo en la respuesta en el tiempo de una aplicación, ya que una capacidad de almacenamiento insuficiente puede ralentizar o incluso impedir que la aplicación funcione correctamente. Por lo tanto, es fundamental evaluar y determinar los requisitos de almacenamiento adecuados para una aplicación para garantizar que cumpla con los requerimientos no funcionales relacionados con la respuesta en el tiempo.

### Respuesta Esperada:

Según el contexto, la capacidad de almacenamiento tiene un impacto directo en la respuesta en el tiempo, ya que es uno de los requerimientos no funcionales que se refiere a las propiedades emergentes del sistema. En otras palabras, la capacidad de almacenamiento influye en la velocidad y eficiencia con la que el sistema responde a las solicitudes.

### Contexto Relevante:

1. Requerimientos no funcionales Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. modelo de despliegue como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones ram procesadores espacio de almacenamiento ancho de banda 1 según los requisitos de la aplic.....

3. **t e s t e a b i l i d a d** se debe diseñar sus componentes para que sean fácilmente probados d e s p l e g a b i l i d a d facilidad con la que se puede pasar de desarrollo a producción e v o l u c i o n r e l e a s.....

### **Métricas de Evaluación:**

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the output mostly addresses the impact of storage capacity on response time, but gets sidetracked by mentioning unrelated details like Document 7 and specific response time requirements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 mentioning 'requerimientos no funcionales' and node 2 mentioning 'espacio de almacenamiento', are ranked higher than the irrelevant node 3, which talks about 't e s t e a b i l i d a d' and is not related to the topic.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the expected output perfectly aligns with the nodes in retrieval context, specifically nodes 1 and 2, which explicitly mention the impact of storage capacity on system response time.

Contextual Relevancy: 0.33 (Threshold: 0.5) - The score is 0.33 because the retrieval context is mostly irrelevant to the input, with statements discussing unrelated topics such as 'restricciones del sistema', 'inteligencia artificial', and 'reglamentos de seguridad', and only a few statements mentioning 'respuesta en el tiempo' and 'capacidad de almacenamiento' in a relevant way.



## Caso de Prueba: 208 - Fallo

### Input del Usuario:

¿Cuál es el factor interno que se refiere a la capacidad de un sistema para trabajar con otros sistemas de software o hardware?

### Respuesta Generada:

Según el Documento 1, el factor interno que se refiere a la capacidad de un sistema para trabajar con otros sistemas de software o hardware es la "necesidad de interoperabilidad".

### Respuesta Esperada:

El factor interno que se refiere a la capacidad de un sistema para trabajar con otros sistemas de software o hardware es la necesidad de interoperabilidad.

### Contexto Relevante:

1. Requerimientos no funcionales Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad.....

### Contexto Recuperado:

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. testabilidad se debe diseñar sus componentes para que sean fácilmente probados de manera que la fiabilidad con la que se puede pasar de desarrollo a producción y viceversa.....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitación es el proceso de recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node which mentions 'necesidad de interoperabilidad con otros sistemas de software o hardware', and the second node which mentions 'interoperabilidad', are ranked higher than the irrelevant node at rank 3, which talks about 'proceso para especificar requerimientos'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, which explicitly mentions the key concept of interoperability.

Contextual Relevancy: 0.10 (Threshold: 0.5) - The score is 0.10 because the retrieval context mostly talks about unrelated concepts like non-functional requirements, system restrictions, and external factors, as stated in the reasons for irrelevancy, and only briefly mentions interoperability in a few statements.

## **Caso de Prueba: 209 - Éxito**

### **Input del Usuario:**

¿Cuáles son los factores internos y externos que influyen en la definición de los requerimientos no funcionales de seguridad en un sistema?

### **Respuesta Generada:**

Según el Documento 1, los factores internos que influyen en la definición de los requerimientos no funcionales de seguridad en un sistema son: \* La necesidad del usuario \* Las restricciones en el presupuesto \* Las políticas de la organización \* La necesidad de interoperabilidad con otros sistemas de software o hardware Por otro lado, los factores externos que influyen en la definición de los requerimientos no funcionales de seguridad en un sistema son: \* Los reglamentos de seguridad \* Las políticas del gobierno \* Otros factores externos no mencionados explícitamente en el documento, pero que pueden influir en la definición de los requerimientos no funcionales de seguridad en un sistema. Es importante destacar que estos factores pueden variar dependiendo del sistema y de la organización que lo desarrolla, por lo que es importante identificar los factores específicos que influyen en la definición de los requerimientos no funcionales de seguridad en un sistema determinado.

### **Respuesta Esperada:**

Los factores internos que influyen en la definición de los requerimientos no funcionales de seguridad en un sistema son la necesidad del usuario, las restricciones en el presupuesto, las políticas de la organización y la necesidad de interoperabilidad con otros sistemas de software o hardware. Los factores externos son los reglamentos de seguridad y las políticas del gobierno, entre otros.

### **Contexto Relevante:**

1. Requerimientos no funcionales Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad.....

### **Contexto Recuperado:**

1. requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este como la fiabilidad.....

2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

3. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a comprehensive and accurate response without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly addressing the question and the irrelevant nodes, such as node 2 discussing functional requirements and node 3 discussing imprecision in requirements, correctly ranked lower.

Contextual Recall: 0.50 (Threshold: 0.5) - The score is 0.50 because the model was able to attribute the first sentence to node 1 in the retrieval context, but failed to find a match for the second sentence.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context has some relevant statements that mention non-functional requirements, such as 'requerimientos no funcionales' and 'factores internos y externos', but is also cluttered with irrelevant information about functional requirements, as stated in the reasons for irrelevancy, e.g., 'The statement talks about functional requirements, which is not related to non-functional security requirements.'

## Caso de Prueba: 210 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre escribir programas y desarrollar software?

### Respuesta Generada:

Hola! La principal diferencia entre escribir programas y desarrollar software se debe a la complejidad y el enfoque en la creación del producto final. Escribir programas se refiere a la creación de código fuente que soluciona un problema específico o automatiza una tarea. El enfoque está en escribir código que funcione correctamente y produzca los resultados esperados. Sin embargo, esta actividad no considera la planificación, el diseño, la pruebas, la implantación y el mantenimiento del sistema como un todo. Por otro lado, desarrollar software implica un proceso más amplio y estructurado que abarca todas las fases del ciclo de vida del desarrollo de software, desde la concepción hasta su puesta en producción. Se busca crear un producto de alta calidad que satisfaga las necesidades y expectativas de los usuarios. El enfoque está en aplicar conceptos, métodos y herramientas para crear aplicaciones de software que sean eficientes, escalables y mantenibles en el tiempo. En resumen, escribir programas se centra en crear código que funcione, mientras que desarrollar software se centra en crear un producto de software completo y de alta calidad que satisfaga las necesidades de los usuarios.

### Respuesta Esperada:

La principal diferencia entre escribir programas y desarrollar software es que escribir programas se enfoca en aplicar conceptos, métodos y herramientas para crear aplicaciones de software, mientras que desarrollar software implica un proceso más amplio que incluye el ciclo de vida, el proceso de creación de software, la elicitud de requerimientos, entre otros aspectos.

### Contexto Relevante:

1. Presentación de curso Características del curso Definiciones Qué es software ¿Qué es ingeniería de software? Aplicar conceptos, métodos y herramientas para crear aplicaciones de software. Diferencias .....

### Contexto Recuperado:

1. presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....

2. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

3. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'aplicar conceptos, metodos y herramientas para crear aplicaciones de software', is relevant to the input, and the third-ranked node, which mentions 'desarrollo de software agil', is also relevant, but the second-ranked node, which mentions 'el software funcionando es la medida principal de progreso', is an irrelevant node that should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless connection between the input and the output.

Contextual Relevancy: 0.19 (Threshold: 0.5) - The score is 0.19 because the retrieval context is mostly filled with irrelevant statements about course presentation, software definitions, and development life cycles, as stated in the reasons for irrelevancy, with only a few phrases like 'diferencias escribir programas vs desarrollar software' and 'desarrollo de software agil resumen' being remotely relevant to the input.

## Caso de Prueba: 211 - Éxito

### Input del Usuario:

¿Cuál es el propósito del ciclo de vida en el proceso de creación de aplicaciones de software?

### Respuesta Generada:

Según los documentos proporcionados, el propósito del ciclo de vida en el proceso de creación de aplicaciones de software es representar las etapas que describen el proceso de desarrollo de software desde su concepción hasta su puesta en producción. En otras palabras, el ciclo de vida del desarrollo de software es un proceso estructurado que guía el desarrollo del software desde su inicio hasta su finalización, pasando por fases como la planificación, análisis, diseño, codificación, pruebas y despliegue. Este ciclo de vida permite organizar y gestionar las tareas involucradas en each fase del desarrollo de software, con el fin de mejorar la calidad del producto final, optimizar los recursos disponibles y minimizar los riesgos asociados al desarrollo. En resumen, el propósito del ciclo de vida en el proceso de creación de aplicaciones de software es proporcionar un marco estructurado para el desarrollo de software, permitiendo la creación de aplicaciones de alta calidad y eficientes.

### Respuesta Esperada:

El propósito del ciclo de vida en el proceso de creación de aplicaciones de software es proporcionar un proceso o marco estructurado para el desarrollo de software.

### Contexto Relevante:

1. Presentación de curso Características del curso Definiciones Qué es software ¿Qué es ingeniería de software? Aplicar conceptos, métodos y herramientas para crear aplicaciones de software. Diferencias .....

### Contexto Recuperado:

1. presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....
2. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....
3. ciclo de vida del desarrollo de software lluvia de ideas que resuelven un problema particular que enfrentan los usuarios ideacion concepto ciclo de vida del software hoy dia interactuar con los stake.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question about the purpose of the software development life cycle, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 in the retrieval contexts directly answers the question, and node 3 is also relevant, but node 2, which is ranked second, is an irrelevant node that explains other concepts, and should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, resulting in a flawless match.

Contextual Relevancy: 0.64 (Threshold: 0.5) - The score is 0.64 because the retrieval context has many irrelevant statements, such as 'presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software', that are not related to the purpose of the software development life cycle, but it still contains some relevant statements like 'el ciclo de vida del desarrollo de software representa las etapas que describen el proceso de desarrollo de software desde su concepcion hasta su puesta en produccion' that are directly related to the input.



## Caso de Prueba: 212 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre escribir programas y desarrollar software?

### Respuesta Generada:

Según el Documento 1, la principal diferencia entre escribir programas y desarrollar software es que desarrollar software implica aplicar conceptos, métodos y herramientas para crear aplicaciones de software, lo que abarca un proceso más amplio que simplemente escribir código. En otras palabras, desarrollar software involucra una serie de etapas, como el análisis, el diseño, la codificación, las pruebas y el despliegue, mientras que escribir programas se enfoca más en la creación de código. En otras palabras, escribir programas se centra en la implementación de una solución mediante código, mientras que desarrollar software implica un enfoque más sistemático y estructurado para crear una aplicación completa, teniendo en cuenta los requisitos del usuario, la planificación, la gestión del proyecto y la colaboración en equipo. Es importante destacar que desarrollar software también implica considerar aspectos como la calidad, la seguridad, la escalabilidad y la mantenibilidad del software, lo que no siempre es considerado cuando simplemente se escribe un programa.

### Respuesta Esperada:

La principal diferencia entre escribir programas y desarrollar software es que escribir programas se enfoca en aplicar conceptos, métodos y herramientas para crear aplicaciones de software, mientras que desarrollar software implica un proceso más amplio que incluye el ciclo de vida, el proceso de creación de software, la elicitation de requerimientos y la aplicación de metodologías, entre otros aspectos.

### Contexto Relevante:

1. Presentación de curso Características del curso Definiciones Qué es software ¿Qué es ingeniería de software? Aplicar conceptos, métodos y herramientas para crear aplicaciones de software. Diferencias .....

### Contexto Recuperado:

1. presentacion de curso caracteristicas del curso definiciones que es software que es ingenieria de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....

2. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

3. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a precise and relevant answer!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'aplicar conceptos, metodos y herramientas para crear aplicaciones de software', is relevant to the input, and the third-ranked node, which mentions 'desarrollo de software agil', is also relevant. However, the second-ranked node, which mentions '07 el software funcionando es la medida principal de progreso...', is not relevant and should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, with all sentences aligning with the node's content.

Contextual Relevancy: 0.19 (Threshold: 0.5) - The score is 0.19 because the retrieval context is largely irrelevant to the input, with most statements discussing unrelated topics such as course presentation, software development life cycle, and agile processes, as quoted in the reasons for irrelevancy, whereas only a few statements like 'diferencias escribir programas vs desarrollar software' and 'desarrollo de software agil resumen' are remotely relevant to the input.

## Caso de Prueba: 213 - Éxito

### Input del Usuario:

¿Cuáles son los tipos de requerimientos en el desarrollo de software?

### Respuesta Generada:

Según los documentos proporcionados, existen dos tipos principales de requerimientos en el desarrollo de software: 1. **\*\*Requerimientos funcionales\*\***: Estos se refieren a las declaraciones de los servicios que prestará el sistema, como la forma en que reaccionará a determinados insumos. Describen el comportamiento del sistema en condiciones específicas y establecen explícitamente lo que el sistema debe hacer. Ejemplos de requerimientos funcionales incluyen "El sistema debe permitir crear un usuario del sistema" o "El sistema debe permitir registrar la venta de un artículo". 2. **\*\*Requerimientos no funcionales\*\***: Estos no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. Definen las restricciones del sistema, como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interfaz del sistema. Ejemplos de requerimientos no funcionales incluyen la seguridad, la interoperabilidad con otros sistemas, o la capacidad de respuesta en tiempo real. Es importante destacar que estos dos tipos de requerimientos son complementarios y se necesitan ambos para desarrollar un sistema de software que cumpla con las necesidades del usuario.

### Respuesta Esperada:

Según el contexto, los tipos de requerimientos en el desarrollo de software son: funcionales y no funcionales.

### Contexto Relevante:

1. Presentación de curso Características del curso Definiciones Qué es software ¿Qué es ingeniería de software? Aplicar conceptos, métodos y herramientas para crear aplicaciones de software. Diferencias .....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. imprecision en los requerimientos los problemas surgen cuando los requerimientos no son declarados con precision requerimientos ambiguos pueden interpretarse de diferentes maneras por los desarrollado.....
3. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question about software development requirements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts, which mentions 'requerimientos no funcionales' and 'requerimientos funcionales', is highly relevant to the input, and the third-ranked node, which explicitly defines 'requerimientos funcionales', is also relevant, but the second-ranked node, which discusses 'imprecision en los requerimientos', is not directly related to the types of requirements, which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes in the retrieval context, specifically nodes 1 and 3, which explicitly mention 'requerimientos funcionales' and 'requerimientos no funcionales'.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly matches the input, providing a comprehensive list of software development requirements, including non-functional, functional, and validation requirements, making it a perfect fit!

## Caso de Prueba: 214 - Éxito

### Input del Usuario:

¿Cuál es la relación entre el software y la ingeniería de software?

### Respuesta Generada:

Según los documentos proporcionados, la relación entre el software y la ingeniería de software es que la ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. En otras palabras, la ingeniería de software es la disciplina que se encarga del diseño, desarrollo, implementación, pruebas y mantenimiento del software. El software, por otro lado, es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que, al ser ejecutadas por un equipo informático (hardware), permiten realizar tareas específicas y procesar información. En resumen, la ingeniería de software es la disciplina que se encarga de crear y mantener el software, que es el producto final que se utiliza para realizar tareas específicas y procesar información.

### Respuesta Esperada:

La relación entre el software y la ingeniería de software es que la ingeniería de software es la disciplina que se encarga del diseño, desarrollo, implementación, pruebas y mantenimiento del software. En otras palabras, la ingeniería de software es el proceso que se utiliza para crear el software.

### Contexto Relevante:

1. 4 ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten reali.....

### Contexto Recuperado:

1. conceptos que es ingenieria de software la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas desplieg.....
2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion que al ser ejecutadas por un equipo informatico hardware le permiten realizar tarea.....
3. aplicar conocimiento la ingenieria de software aplica el conocimiento y la comprension teorica obtenidos a traves de la computacion para la creacion de aplicaciones de software de calidad a lta d e m.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a precise and relevant answer to the question about the relationship between software and software engineering!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the top two nodes directly addressing the relationship between software and software engineering, and the third node, ranked lowest, only being related to software engineering but not directly addressing the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly defines software engineering as a discipline that applies principles, methodologies, and practices to software design, development, implementation, testing, and maintenance.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context provides relevant statements that directly address the relationship between software and software engineering, such as 'la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al disenno desarrollo implementacion pruebas despliegue monitoreo y mantenimiento de software' and 'el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion', which outweigh the irrelevant statements.

## Caso de Prueba: 215 - Éxito

### Input del Usuario:

¿Cuál es la relación entre el lenguaje de programación y la ingeniería de software?

### Respuesta Generada:

Según los documentos proporcionados, la relación entre el lenguaje de programación y la ingeniería de software es que el lenguaje de programación es una herramienta utilizada en el proceso de desarrollo de software, que es una parte fundamental de la ingeniería de software. En específico, el lenguaje de programación se utiliza en la codificación, que es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación (Documento 5). La ingeniería de software, por otro lado, se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios, y abarca todo el ciclo de vida del desarrollo de software, desde su concepción hasta su puesta en producción (Documentos 1, 4 y 6). En otras palabras, la programación es un proceso que se encuentra dentro del contexto más amplio de la ingeniería de software, y es una de las etapas clave en la creación de software. La ingeniería de software, por su parte, se ocupa de la planificación, el diseño, la implementación, las pruebas, el despliegue, el monitoreo y el mantenimiento de software, y utiliza el lenguaje de programación como una herramienta para lograr estos objetivos. En resumen, la relación entre el lenguaje de programación y la ingeniería de software es que el lenguaje de programación es un instrumento utilizado en la programación, que es una parte del proceso de desarrollo de software, que a su vez es una parte fundamental de la ingeniería de software.

### Respuesta Esperada:

La relación entre el lenguaje de programación y la ingeniería de software es que el lenguaje de programación es una herramienta utilizada en la ingeniería de software para codificar instrucciones que, al ser ejecutadas por un equipo informático, permiten realizar tareas específicas y procesar información. La ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad, y el lenguaje de programación es un elemento fundamental en este proceso.

### Contexto Relevante:

1. 4 ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten reali.....

### Contexto Recuperado:

1. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....
2. conceptos que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un sistema informático hardware le permiten rea.....
3. conceptos programación vs desarrollo de software la programación es el proceso de trasladar un problema de su ambiente físico a un lenguaje que el computador pueda entender y obedecer uno o dos

desarr.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a direct and relevant response without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which explains the relationship between programming language and software engineering, and node 2 which provides additional details about software functionality and performance, are ranked higher than the irrelevant node 3 which only discusses the difference between programming and software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which provides a clear and comprehensive explanation of the relationship between programming language and software engineering.

Contextual Relevancy: 0.71 (Threshold: 0.5) - The score is 0.71 because the retrieval context is only partially relevant, as it contains some relevant statements like 'el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programacion' and 'la ingenieria de software es una disciplina que aplica principios metodologias y practicas de la ingenieria al diseno desarrollo implementacion pruebas y mantenimiento de software', but also includes irrelevant information like 'satisfaga las necesidades y expectativas de los usuarios' and 'characteristics of small applications'.



## Caso de Prueba: 216 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de la ingeniería de software?

### Respuesta Generada:

De acuerdo con los Documentos 1 y 2, el propósito principal de la ingeniería de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. La ingeniería de software aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas, despliegue, monitoreo y mantenimiento de software.

### Respuesta Esperada:

El propósito principal de la ingeniería de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Contexto Relevante:

1. 4 ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten reali.....

### Contexto Recuperado:

1. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

2. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

3. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant answer to the question about the main purpose of software engineering!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which directly answer the question, are ranked higher than the irrelevant nodes, like node 3, which does not directly address the main purpose of software engineering.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st and 2nd nodes in the retrieval context, showcasing a seamless alignment between the two!

Contextual Relevancy: 0.58 (Threshold: 0.5) - The score is 0.58 because the retrieval context only partially addresses the main purpose of software engineering, with relevant statements like 'la ingenieria de software es una disciplina...' and 'se enfoca en la creacion sistematica...', but is largely irrelevant due to reasons such as 'The statement is about the focus of software engineering, not its main purpose' and 'The retrieval context contained unrelated information'.

## Caso de Prueba: 217 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de la ingeniería de software?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal de la ingeniería de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. (Documento 1 y Documento 2) La ingeniería de software aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas y mantenimiento de software, con el objetivo de crear software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Respuesta Esperada:

El propósito principal de la ingeniería de software es la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

### Contexto Relevante:

1. 4 ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten reali.....

### Contexto Recuperado:

1. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

2. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

3. 07 el software funcionando es la medida principal de progreso los procesos ágiles promueven el desarrollo sostenible la atención continua a la excelencia técnica y al buen diseño mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addressed the input question about the main purpose of software engineering!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which is about agile development and does not directly address the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes 1 and 2 in the retrieval context, showcasing a seamless alignment.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context has some relevant statements that define software engineering and its focus on creating high-quality software, but is brought down by irrelevant statements that are too broad, vague, or unrelated to the main purpose of software engineering, such as '07 el software funcionando es la medida principal de progreso' and 'la simplicidad o el arte de maximizar la cantidad de trabajo realizado es esencial'.

## Caso de Prueba: 218 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de utilizar la técnica de lluvia de ideas en la diagramación de requerimientos no funcionales?

### Respuesta Generada:

Según el Documento 1, el propósito principal de utilizar la técnica de lluvia de ideas en la diagramación de requerimientos no funcionales es "identificar posibles soluciones a los problemas" y "proponer varias alternativas" para resolver un problema particular. En otras palabras, la lluvia de ideas se utiliza para generar y recopilar ideas y soluciones posibles para un problema o necesidad específica, lo que ayuda a clarificar las cosas y a capturar la colaboración y las ideas de los participantes.

### Respuesta Esperada:

El propósito principal de utilizar la técnica de lluvia de ideas en la diagramación de requerimientos no funcionales es identificar posibles soluciones a los problemas, especialmente en aplicaciones nuevas donde hay pocos antecedentes.

### Contexto Relevante:

1. Requerimientos no funcionales Técnicas para obtener requerimientos Lluvia de ideas: Reunión donde se proponen ideas para solucionar un problema. • Tener un plan : dividir el problema en partes pequeña.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para obtener requerimientos Lluvia de ideas reunion donde se proponen ideas para solucionar un problema tener un plan dividir el problema en partes pequenas ej .....

2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....

3. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....

### Métricas de Evaluación:

Answer Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the response is mostly relevant, but slightly derailed by mentioning a secondary benefit, which prevented it from being a perfect match.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which doesn't mention the purpose of brainstorming, and node 3 which doesn't specifically address the purpose, are correctly ranked lower than the relevant node 1, which clearly addresses the

question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions the purpose of using brainstorming techniques for identifying possible solutions in new applications with few precedents.

Contextual Relevancy: 0.41 (Threshold: 0.5) - The score is 0.41 because the retrieval context mostly contains irrelevant statements, with 15 out of 17 statements not related to the purpose of brainstorming in non-functional requirements diagramming, as they discuss other topics such as validation, prototyping, and documentation.

## Caso de Prueba: 219 - Éxito

### Input del Usuario:

¿Cuál es la relación entre el software y la ingeniería de software?

### Respuesta Generada:

Según los documentos proporcionados, la relación entre el software y la ingeniería de software es la siguiente: El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que, al ser ejecutadas por un equipo informático (hardware), le permiten realizar tareas específicas y procesar información (Documento 2). La ingeniería de software, por otro lado, es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas y mantenimiento de software (Documentos 1 y 2). La ingeniería de software se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios. En otras palabras, el software es el producto que se desarrolla, mientras que la ingeniería de software es la disciplina que se ocupa de diseñar, desarrollar, implementar y mantener ese software de manera eficiente y de alta calidad. La ingeniería de software utiliza conocimientos y comprensión teóricos obtenidos a través de la computación para crear aplicaciones de software de calidad (Documento 3). En resumen, la relación entre el software y la ingeniería de software es que el software es el objeto de estudio y desarrollo, mientras que la ingeniería de software es la disciplina que se ocupa de diseñar, desarrollar y mantener ese software de manera eficiente y efectiva.

### Respuesta Esperada:

La relación entre el software y la ingeniería de software es que la ingeniería de software se enfoca en el diseño, desarrollo, implementación, pruebas y mantenimiento del software. En otras palabras, la ingeniería de software es el proceso que se utiliza para crear el software.

### Contexto Relevante:

1. 4 ¿Qué es software ? El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten reali.....

### Contexto Recuperado:

1. conceptos que es ingeniería de software la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas desplieg.....

2. que es software el software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación que al ser ejecutadas por un equipo informático hardware le permiten realizar tarea.....

3. aplicar conocimiento la ingeniería de software aplica el conocimiento y la comprensión teórica obtenidos a través de la computación para la creación de aplicaciones de software de calidad a l t a d e m.....

## **Métricas de Evaluación:**

Answer Relevancy: 0.62 (Threshold: 0.5) - The score is 0.62 because the output partially addresses the input question, but lacks clear explanations on the relation between software and software engineering, getting sidetracked by defining software and its functionality.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes in the retrieval contexts, such as node 1 and node 2, are correctly ranked higher than the irrelevant nodes, ensuring that the most accurate information is prioritized.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly outlines the software engineering process.

Contextual Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the retrieval context is highly relevant to the input, as it provides direct definitions and explanations of software engineering, such as 'la ingeniería de software es una disciplina que aplica principios metodologías y prácticas de la ingeniería al diseño desarrollo implementación pruebas despliegue monitoreo y mantenimiento de software', which aligns perfectly with the input question.



## Caso de Prueba: 220 - Fallo

### Input del Usuario:

¿Cuál es la relación entre el análisis y el diseño en el ciclo de vida del desarrollo de software?

### Respuesta Generada:

Según los contextos proporcionados, especialmente el Documento 1 y Documento 5, se puede concluir que el análisis y el diseño son dos etapas fundamentales en el ciclo de vida del desarrollo de software. El análisis es el proceso de entender el problema o necesidad de los usuarios y, a partir de esta comprensión, se escriben los requerimientos del sistema. En otras palabras, el análisis implica la recolección y la documentación de los requisitos del sistema, tanto funcionales como no funcionales. Por otro lado, el diseño es la descripción o especificación de los procesos, componentes e interfaces que representan una solución para el sistema a construir. En este sentido, el diseño toma los requisitos identificados en el análisis y los transforma en una especificación detallada de cómo se va a implementar el sistema. La relación entre el análisis y el diseño es, por lo tanto, de causa y efecto. El análisis proporciona los requisitos que deben ser satisfechos por el sistema, y el diseño es la respuesta a esos requisitos, es decir, la especificación de cómo se va a construir el sistema para satisfacer esas necesidades. En resumen, el análisis es el proceso de comprender el problema o necesidad, mientras que el diseño es el proceso de especificar la solución. Ambas etapas son fundamentales en el ciclo de vida del desarrollo de software y están estrechamente relacionadas.

### Respuesta Esperada:

La relación entre el análisis y el diseño en el ciclo de vida del desarrollo de software es que el análisis proporciona los requerimientos del sistema, y el diseño utiliza estos requerimientos para describir o especificar la solución del sistema a construir. En otras palabras, el análisis define el problema y el diseño ofrece una solución para ese problema.

### Contexto Relevante:

1. 6 Conceptos Análisis: Es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema Diseño: Es una descripción o especificación de pro.....

### Contexto Recuperado:

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

2. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....

3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant response to the question about the relationship between analysis and design in the software development life cycle!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as the first node that clearly explains the relationship between analysis and design, are ranked higher than irrelevant nodes, which do not directly address the relationship, like the second and third nodes.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly defines the analysis and design process in software development.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because most of the retrieval context statements, such as 'aspectos conceptuales del world wide web', 'el rol de los servidores y clientes web', and 'peticiones get y post', are not related to the relationship between analysis and design in the software development life cycle, and only a few statements, like 'conceptos analisis es el proceso de entender el problema...' and 'diseno es una descripcion...', are partially relevant.

## Caso de Prueba: 221 - Fallo

### Input del Usuario:

¿Cuál es la relación entre el análisis y el diseño en el ciclo de vida del desarrollo de software?

### Respuesta Generada:

Según los conceptos presentados en los documentos proporcionados, el análisis y el diseño son dos etapas consecutivas y relacionadas en el ciclo de vida del desarrollo de software. El análisis se refiere al proceso de entender el problema o necesidad de los usuarios y recopilar los requisitos del sistema. En este sentido, el análisis implica la recolección de información, la identificación de los requisitos funcionales y no funcionales, y la documentación de los mismos. Por otro lado, el diseño se refiere a la descripción o especificación de los procesos, componentes e interfaces que representan una solución para el sistema a construir. El diseño toma como entrada los requisitos recopilados durante el análisis y utiliza esta información para crear una representación abstracta del sistema que se va a desarrollar. En otras palabras, el análisis proporciona la entrada necesaria para el diseño, ya que los requisitos recopilados en el análisis se utilizan para crear la arquitectura del sistema, sus componentes, interfaces y forma de interactuar. De esta manera, el análisis y el diseño están estrechamente relacionados, ya que el diseño se basa en la información recopilada durante el análisis. En resumen, la relación entre el análisis y el diseño en el ciclo de vida del desarrollo de software es que el análisis proporciona la entrada necesaria para el diseño, y el diseño utiliza esta información para crear una solución para el sistema a construir.

### Respuesta Esperada:

La relación entre el análisis y el diseño en el ciclo de vida del desarrollo de software es que el análisis proporciona los requerimientos del sistema, y el diseño utiliza estos requerimientos para describir o especificar la solución para el sistema a construir. En otras palabras, el análisis define el problema y el diseño propone la solución.

### Contexto Relevante:

1. 6 Conceptos Análisis: Es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema Diseño: Es una descripción o especificación de pro.....

### Contexto Recuperado:

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

2. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....

3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

## **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a direct and accurate response to the question about the relationship between analysis and design in the software development life cycle!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node, which clearly explains the relationship between analysis and design, ranked highest, and the subsequent nodes, which do not directly address the relationship, ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the nodes in retrieval context, specifically nodes 1, which defines the analysis and design concepts, accurately reflecting the relationship between them.

Contextual Relevancy: 0.44 (Threshold: 0.5) - The score is 0.44 because the retrieval context contains mostly irrelevant information, such as agile methodologies, traditional methodologies, and web development concepts, which 'is not directly related to the relationship between analysis and design in the software development life cycle', and only a few relevant statements that briefly touch on the topic.

## Caso de Prueba: 222 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal del análisis en el ciclo de vida del desarrollo?

### Respuesta Generada:

Según el contexto proporcionado, específicamente en el Documento 1, el objetivo principal del análisis en el ciclo de vida del desarrollo es entender el problema o necesidad de los usuarios y, a partir de esas necesidades, escribir los requerimientos de un sistema. En otras palabras, el análisis es el proceso de comprender los problemas o necesidades de los usuarios para luego definir los requisitos del sistema que se va a desarrollar.

### Respuesta Esperada:

El objetivo principal del análisis en el ciclo de vida del desarrollo es entender el problema o necesidad de los usuarios y escribir los requerimientos de un sistema.

### Contexto Relevante:

1. 6 Conceptos Análisis: Es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema Diseño: Es una descripción o especificación de pro.....

### Contexto Recuperado:

1. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....
2. aspectos a tratar conceptos generales el proceso de desarrollo de software ciclo de vida de desarrollo del software modelos de ciclo de vida del software conceptos sobre metodologias de desarrollo de .....
3. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos metodos y herramientas para crear aplicaciones de software diferencias escr.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly aligns with the input, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the first node directly addressing the question and irrelevant nodes, such as the second node discussing general software development topics and the third node being a course introduction, ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output perfectly matches the 1st node in the retrieval context, showcasing a seamless understanding of the development cycle's analysis phase.

Contextual Relevancy: 0.17 (Threshold: 0.5) - The score is 0.17 because the retrieval context is largely irrelevant, with most statements having 'nothing to do with the main objective of analysis in the software development life cycle', as stated in the reasons for irrelevancy, and only a few statements, such as 'el ciclo de vida del desarrollo de software representa las etapas que describen el proceso de desarrollo de software desde su concepcion hasta su puesta en produccion', being somewhat relevant.

## Caso de Prueba: 223 - Fallo

### Input del Usuario:

¿Cuál es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación?

### Respuesta Generada:

Según el Documento 1, el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación se conoce como "codificación". La codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación.

### Respuesta Esperada:

La respuesta es: La codificación.

### Contexto Relevante:

1. 7 Conceptos La codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación Las pruebas de software es el proceso de evaluar y verificar que un sistema o.....

### Contexto Recuperado:

1. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

2. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

3. ejercicio elaborar la descripcion de un requerimiento funcional para el proceso de registro de un curso en el sistema de la universidad el sistema debe permitir crear un usuario del sistema...

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addressed the input question without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 which talks about 'conceptos analisis' and node 3 which is an exercise, are correctly ranked lower than the highly relevant node 1 that directly answers the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in the retrieval context, indicating a flawless recall!

Contextual Relevancy: 0.20 (Threshold: 0.5) - The score is 0.20 because the retrieval context mostly talks about software testing, development life cycle, and user needs, which are not directly related to transforming a system design into a programming language format, as stated in reasons for irrelevancy, but has a few relevant statements like 'conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion' that slightly increase the score.



## Caso de Prueba: 224 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de las pruebas de software en el ciclo de vida del desarrollo de software?

### Respuesta Generada:

De acuerdo con el contexto proporcionado, el propósito principal de las pruebas de software en el ciclo de vida del desarrollo de software es evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer. En otras palabras, las pruebas de software tienen como objetivo asegurarse de que el software cumpla con los requisitos y expectativas de los usuarios, es decir, que funcione correctamente y según lo diseñado.

### Respuesta Esperada:

El propósito principal de las pruebas de software en el ciclo de vida del desarrollo de software es evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer.

### Contexto Relevante:

1. 7 Conceptos La codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación Las pruebas de software es el proceso de evaluar y verificar que un sistema o.....

### Contexto Recuperado:

1. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

2. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....

3. conceptos metodologías de desarrollo de software una metodología de desarrollo de software es un conjunto estructurado de prácticas principios y procedimientos que guían y regulan el proceso de creaci.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a direct and accurate response to the question!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as node 2 with the reason "'presentacion de curso características del curso...' is not directly related to the purpose of software testing in the software development life cycle." and node 3 with the reason "'conceptos metodologías de desarrollo de software...' talks about software development

methodologies, but does not explicitly mention the purpose of software testing.", are correctly ranked lower than the relevant node 1.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly matched with the 2nd node in the retrieval context, resulting in a flawless alignment.

Contextual Relevancy: 0.06 (Threshold: 0.5) - The score is 0.06 because the retrieval context is mostly irrelevant, with many statements not mentioning the purpose of software testing in the software development life cycle, as stated in the reasons for irrelevancy, and only one statement being relevant, which is 'las pruebas de software es el proceso de evaluar y verificar que un sistema o aplicacion de software hace lo que se supone que debe hacer'.

## Caso de Prueba: 225 - Éxito

### Input del Usuario:

¿Cuáles son las características que debe tener un requisito del software para ser considerado válido?

### Respuesta Generada:

Según los documentos proporcionados, las características que debe tener un requisito del software para ser considerado válido son: 1. **\*\*Necesario\*\***: Lo que pida un requisito debe ser necesario para el producto correcto. 2. **\*\*No ambiguo\*\***: El texto debe ser claro, preciso y tener una única interpretación posible. 3. **\*\*Conciso\*\***: Debe redactarse en un lenguaje comprensible por los clientes, en lugar de uno de tipo técnico y especializado. 4. **\*\*Consistente\*\***: Ningún requisito debe entrar en conflicto con otro requisito diferente, y el lenguaje empleado entre los diferentes requisitos debe ser consistente. 5. **\*\*Completo\*\***: Los requisitos deben contener en sí mismos toda la información necesaria y no remitir a otras fuentes externas que los expliquen con más detalle. 6. **\*\*Alcanzable\*\***: Un requisito debe ser un objetivo realista, posible de ser alcanzado en el tiempo y los recursos disponibles. 7. **\*\*Verificable\*\***: Se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no, mediante inspección, análisis, demostración o testeo. Es importante destacar que un requisito del software debe cumplir con estas características para ser considerado válido y, por lo tanto, para ser incluido en el desarrollo del software.

### Respuesta Esperada:

Según el contexto, un requisito del software debe tener las siguientes características para ser considerado válido: \* No ambiguo \* Comprobable \* Medible \* Necesario para la aceptación del producto o proceso

### Contexto Relevante:

1. 8 REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
2. detalles de una historia de usuario criterios de aceptación item criterio de aceptación las condiciones que un producto de software debe satisfacer para ser aceptado por un usuario o cliente un usuario.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise answer without any irrelevant information!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node in the retrieval contexts correctly addresses the input question, and the third-ranked node also aligns with the expected output. However, the second-ranked node is an irrelevant node, as it discusses 'detalles de una historia de usuario', which is not directly related to the characteristics of a valid software requirement, and should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly supported by nodes in the retrieval context, specifically nodes 1 and 3, which explicitly mention the required characteristics of a software requirement.

Contextual Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the retrieval context contains many irrelevant test cases and user stories, but also includes several relevant statements that describe characteristics of a valid software requirement, such as being 'no ambiguo', 'conciso', 'consistente', 'completo', 'alcanzable', and 'verificable'.

## Caso de Prueba: 226 - Fallo

### Input del Usuario:

¿Cuáles son las características clave de un requisito en términos de su claridad y comprobabilidad?

### Respuesta Generada:

Según los documentos proporcionados, las características clave de un requisito en términos de su claridad y comprobabilidad son: \* Necesario: El requisito debe ser necesario para el producto correcto. \* No ambiguo: El texto debe ser claro y tener una única interpretación posible. \* Comprobable o medible: Debe ser posible verificar con absoluta certeza si el requisito fue satisfecho o no. Esta verificación puede lograrse mediante inspección, análisis, demostración o testing. \* Consistente: Ningún requisito debe entrar en conflicto con otro requisito diferente. Estas características se mencionan en los Documentos 2, 3 y 4, que discuten la importancia de la claridad y comprobabilidad en la definición de requisitos para el desarrollo de software.

### Respuesta Esperada:

Las características clave de un requisito en términos de su claridad y comprobabilidad son que sea no ambiguo, comprobable y medible.

### Contexto Relevante:

1. 8 REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. tareas investigar que son variables de sesion en una aplicacion web las variables de sesion en una aplicacion web son una forma de almacenar informacion del usuario durante la interaccion con la aplic.....
2. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....
3. requerimientos del software un requisito o requerimiento es una declaración que identifica una característica o restricción operativa funcional o de diseño de un producto o proceso que es no ambigua c.....

### Métricas de Evaluación:

Answer Relevancy: 0.43 (Threshold: 0.5) - The score is 0.43 because the actual output partially addresses the input question, but is often sidetracked by discussing necessity, consistency, and importance, rather than focusing solely on the clarity and verifiability of a requirement.

Contextual Precision: 0.58 (Threshold: 0.5) - The score is 0.58 because the first node, which is irrelevant to the topic of requirements' clarity and verifiability, is ranked higher than the subsequent

nodes that clearly address the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in retrieval context, resulting in a flawless recall.

Contextual Relevancy: 0.47 (Threshold: 0.5) - The score is 0.47 because the retrieval context has many irrelevant statements about session variables, software requirements, and summaries, but only a few relevant statements that actually describe the characteristics of a requirement in terms of clarity and verifiability, such as 'el texto debe ser claro preciso y tener una unica interpretacion posible' and 'se debe poder verificar con absoluta certeza si el requisito fue satisfecho o no'.

## Caso de Prueba: 227 - Fallo

### Input del Usuario:

¿Cuál es la restricción de diseño en un requisito de software?

### Respuesta Generada:

Según los Documentos 1, 2 y 5, un requisito de software es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso. En este sentido, la restricción de diseño se refiere a una limitación o condición específica que se establece en el diseño del software y que debe ser cumplida para que el producto o proceso sea aceptable. En otras palabras, la restricción de diseño es una característica o limitación que se establece en el diseño del software y que debe ser tenida en cuenta durante el desarrollo del mismo. Esto puede incluir, por ejemplo, restricciones en cuanto a la performance, la seguridad, la compatibilidad con otros sistemas, etc. Es importante destacar que, según el Documento 5, un requisito debe ser necesario, no ambiguo, conciso, consistente, completo, alcanzable y verificable. Esto significa que la restricción de diseño debe ser claramente definida, no debe tener múltiples interpretaciones posibles y debe ser posible de implementar y verificar.

### Respuesta Esperada:

La restricción de diseño en un requisito de software es una característica o restricción de diseño de un producto o proceso.

### Contexto Relevante:

1. 8 REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccio operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

3. conceptos la codificacion es el proceso de transformar el diseno de un sistema en un formato de lenguaje de programacion las pruebas de software es el proceso de evaluar y verificar que un sistema o a.....

### Métricas de Evaluación:

Answer Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the output provides some relevant information about software requirements, but it doesn't specifically address the design constraint,

getting sidetracked by general characteristics of requirements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1 and node 2, are correctly ranked higher than irrelevant nodes, like node 3 which talks about 'codificacion' and is not related to design restrictions in software requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches with the 1st node in the retrieval context, which confirms the design restriction in software requirement.

Contextual Relevancy: 0.31 (Threshold: 0.5) - The score is 0.31 because the retrieval context is mostly irrelevant, with statements discussing functional characteristics, software testing, and development life cycle, as stated in the reasons for irrelevancy, such as 'The statement is irrelevant because it talks about functional or operational characteristics, not design restrictions.' and 'The statement explains the software development life cycle and does not relate to software design restrictions.'



## Caso de Prueba: 228 - Éxito

### Input del Usuario:

¿Qué es necesario para la aceptación del producto o proceso según los requisitos del software?

### Respuesta Generada:

Según el Documento 1, para la aceptación del producto o proceso, es necesario que los requisitos del software sean "no ambiguos, comprobables o medibles". Esto significa que los requisitos deben ser claros, precisos y tener una única interpretación posible, de manera que se puedan verificar y cumplir de forma efectiva.

### Respuesta Esperada:

Según los requisitos del software, para la aceptación del producto o proceso es necesario que cumplan con una característica o restricción operativa, funcional o de diseño que sea no ambigua, comprobable o medible.

### Contexto Relevante:

1. 8 REQUERIMIENTOS DEL SOFTWARE Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambig.....

### Contexto Recuperado:

1. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

2. requerimientos del software un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua c.....

3. características de los requerimientos necesario lo que pida un requisito debe ser necesario para el producto correcto si y solo si cada requisito especificado es un requisito que el software debe hacer.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts, ranked 1 to 3, directly address the input question, providing relevant information about the necessary characteristics for acceptance, with the first node directly stating the requirements, the second node reiterating the importance, and the third node adding details on verifiability.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which mirrors the same requirement for product acceptance.

Contextual Relevancy: 0.95 (Threshold: 0.5) - The score is 0.95 because the retrieval context provides highly relevant information about software requirements, such as 'un requisito o requerimiento es una declaracion que identifica una caracteristica o restriccion operativa funcional o de diseno de un producto o proceso que es no ambigua comprobable o medible y necesaria para la aceptacion del producto o proceso', which directly answers the input question about what is necessary for the acceptance of a product or process according to software requirements.

## Caso de Prueba: 229 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de la actividad de identificación de actores en el proceso de elicitar requerimientos?

### Respuesta Generada:

Según los documentos proporcionados, el propósito principal de la actividad de identificación de actores en el proceso de elicitar requerimientos es identificar y conocer a los interesados clave que participan en el sistema de software, como usuarios finales, clientes y otros actores clave. En específico, según el Documento 3, la identificación de actores es parte del proceso de elicitar requerimientos, que tiene como objetivo recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados. Al identificar a los actores, se puede entender mejor las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. En resumen, el propósito principal de la actividad de identificación de actores es identificar a los interesados clave que participan en el sistema de software y entender sus necesidades y expectativas para determinar los requerimientos del sistema.

### Respuesta Esperada:

El propósito principal de la actividad de identificación de actores en el proceso de elicitar requerimientos es identificar a las personas que interactúan con el sistema.

### Contexto Relevante:

1. 13 Proceso para especificar requerimientos Actividades Obtención Requerimientos Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que el siste.....

### Contexto Recuperado:

1. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

2. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 0.71 (Threshold: 0.5) - The score is 0.71 because the response is mostly relevant, but slightly deviates from the specific purpose of identifying actors by discussing the overall goal of the

elicitation process and its achievement.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the top 3 nodes explicitly mentioning 'identificacion de actores' and its relation to the elicitation process, directly aligning with the expected output.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes 1 and 2 in the retrieval context, which explicitly mention key concepts.

Contextual Relevancy: 0.53 (Threshold: 0.5) - The score is 0.53 because the retrieval context is mostly irrelevant to the input, with phrases like 'personas otro sistema dispositivos', 'lo que se desea que el sistema haga', and 'resumen' not directly related to the purpose of identifying actors in the elicitation process, but has some relevant statements like 'identificacion de actores' and 'elicitacion' that slightly improve the score.

## Caso de Prueba: 230 - Éxito

### Input del Usuario:

¿Quiénes son los actores clave que interactúan con el sistema y qué roles juegan en la obtención de requerimientos?

### Respuesta Generada:

Según el Documento 3 y el Documento 4, los actores clave que interactúan con el sistema y juegan un papel importante en la obtención de requerimientos son: \* Personas (usuarios finales, clientes, usuarios contratistas, etc.) \* Otros sistemas \* Dispositivos. Estos actores clave participan en el proceso de elicitación, que consiste en recopilar y descubrir los requerimientos del sistema de software a través de la comunicación directa. El objetivo es identificar las necesidades y expectativas de los usuarios y otras partes interesadas para determinar los requerimientos del sistema. En particular, los usuarios finales y clientes pueden proporcionar información valiosa sobre cómo desean que el sistema funcione y qué funcionalidades deben tener. Los usuarios contratistas, por ejemplo, pueden tener requerimientos específicos para registrar sus pagos de certificación al sistema de seguridad social. Los otros sistemas y dispositivos también pueden tener requerimientos de interfaz y compatibilidad con el sistema. En resumen, estos actores clave interactúan con el sistema y juegan un papel importante en la obtención de requerimientos, ya que proporcionan información valiosa sobre las necesidades y expectativas del sistema de software.

### Respuesta Esperada:

Según el contexto, los actores clave que interactúan con el sistema y juegan un rol en la obtención de requerimientos son: \* Personas \* Otro sistema \* Dispositivos

### Contexto Relevante:

1. 13 Proceso para especificar requerimientos Actividades Obtención Requerimientos Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que el siste.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. requerimientos funcionales los requisitos o requerimientos funcionales son declaraciones de los servicios que prestara el sistema en la forma en que reaccionara a determinados insumos generalmente los.....

3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a precise and relevant answer without any irrelevant statements!

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the first and third nodes in the retrieval contexts, which mention 'comunicacion directa con los interesados' and 'personas, otro sistema, and dispositivos' respectively, are correctly ranked higher than the second node, which only talks about 'requerimientos funcionales' and lacks information about key actors.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, with 'actores clave' in sentence 1 attributed to node 1, 'Personas' and 'Otro sistema' in sentence 2 attributed to node 3, and 'Dispositivos' in sentence 3 also attributed to node 3.

Contextual Relevancy: 0.67 (Threshold: 0.5) - The score is 0.67 because the retrieval context is partially relevant, as it mentions 'elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunicacion directa con los interesados como usuarios finales clientes y otros actores clave' and 'identificacion de actores', which are related to the input, but other statements in the context are not directly related to the key actors and their roles in obtaining requirements, as stated in the reasons for irrelevancy.

## Caso de Prueba: 231 - Fallo

### Input del Usuario:

¿Cuál es el objetivo principal de la actividad de identificación de funcionalidades en el proceso de obtener requerimientos?

### Respuesta Generada:

Según los documentos proporcionados, el objetivo principal de la actividad de identificación de funcionalidades en el proceso de obtener requerimientos es determinar las funcionalidades que debe tener el sistema de software. Esto se logra a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave, con el fin de entender las necesidades y expectativas de los usuarios y otras partes interesadas. En particular, según el Documento 4, la identificación de funcionalidades es parte del proceso de especificación de requerimientos, que incluye la elicitación, análisis y especificación de los requisitos del sistema de software. En este sentido, la identificación de funcionalidades tiene como objetivo definir qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. En resumen, el objetivo principal de la identificación de funcionalidades es determinar qué características y funcionalidades debe tener el sistema de software para cumplir con los objetivos del proyecto y satisfacer las necesidades del cliente.

### Respuesta Esperada:

El objetivo principal de la actividad de identificación de funcionalidades en el proceso de obtener requerimientos es determinar "Lo que se desea que el sistema haga".

### Contexto Relevante:

1. 13 Proceso para especificar requerimientos Actividades Obtención Requerimientos Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que el siste.....

### Contexto Recuperado:

1. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

2. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....

3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because while the output addresses the main objective of identifying functionalities, it gets sidetracked with statements about communication with stakeholders, the overall process of specifying requirements, and meeting project objectives, which dilutes its focus.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 is relevant, but the second node is an irrelevant node, ranked higher than a relevant node, due to 'entender claramente cual es el objetivo de un proyecto...' not being directly related to the identification of funcionalidades.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the 3rd node in retrieval context, which explicitly mentions the key requirement.

Contextual Relevancy: 0.28 (Threshold: 0.5) - The score is 0.28 because most of the retrieval context statements, such as 'identificacion de actores', 'personas otro sistema dispositivos', and 'lo que se desea que el sistema haga', are not directly related to the objective of identifying functional requirements, whereas only a few statements like 'identificacion de funcionalidades' and 'proceso de obtener requerimientos' are relevant.



## Caso de Prueba: 232 - Éxito

### Input del Usuario:

¿Cuál es el propósito de la identificación de actores en el proceso de obtención de requerimientos?

### Respuesta Generada:

Según los documentos proporcionados, el propósito de la identificación de actores en el proceso de obtención de requerimientos es determinar quiénes son los interesados clave que estarán involucrados en el sistema de software, tales como usuarios finales, clientes y otros actores clave. Esto permite recopilar y descubrir los requerimientos del sistema de software a través de la comunicación directa con estos actores, lo que a su vez ayuda a identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para el sistema de software. En otras palabras, la identificación de actores es un paso importante en el proceso de obtención de requerimientos, ya que permite entender quiénes son los usuarios y partes interesadas que interactuarán con el sistema, y qué necesidades y expectativas tienen en cuanto al sistema. Esta información se utilizará posteriormente para definir las funcionalidades y características del software, así como sus restricciones y condiciones de operación.

### Respuesta Esperada:

El propósito de la identificación de actores en el proceso de obtención de requerimientos es determinar quiénes son las personas, sistemas o dispositivos que interactúan con el sistema y qué funcionalidades esperan de él.

### Contexto Relevante:

1. 13 Proceso para especificar requerimientos Actividades Obtención Requerimientos Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que el siste.....

### Contexto Recuperado:

1. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

2. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the top 3 nodes directly addressing the question about the purpose of identifying actors in the process of obtaining requirements, as seen in node 1 where 'identificacion de actores' is mentioned, node 2 where it is part of specifying requirements, and node 3 where it is used to 'determinar los requerimientos'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st and 2nd nodes in the retrieval context, which explicitly mention the key concepts of identifying actors and their expected functionalities.

Contextual Relevancy: 0.59 (Threshold: 0.5) - The score is 0.59 because the retrieval context contains some relevant terms like 'identificacion de actores' and 'proceso de obtencion de requerimientos', but is largely irrelevant due to mentions of unrelated concepts like 'identificacion funcionalidades', 'personas otro sistema dispositivos', and 'lo que se desea que el sistema haga'.

## Caso de Prueba: 233 - Éxito

### Input del Usuario:

¿Cuál es el propósito de la identificación de actores en el proceso de especificación de requerimientos?

### Respuesta Generada:

Según el contexto proporcionado, el propósito de la identificación de actores en el proceso de especificación de requerimientos es determinar quiénes son los interesados clave que intervienen en el sistema de software, como usuarios finales, clientes y otros actores clave. Esto se logra a través de la comunicación directa con estos actores para recopilar y descubrir los requerimientos del sistema de software. En específico, la identificación de actores se menciona en varios documentos, como el Documento 2, Documento 3 y Documento 4, como una actividad dentro del proceso de especificación de requerimientos. Esto sugiere que la identificación de actores es un paso crucial para entender las necesidades y expectativas de los usuarios y otras partes interesadas en el sistema de software. Al identificar a los actores, se puede determinar quiénes serán los usuarios del sistema, qué roles desempeñarán y qué funcionalidades y características serán necesarias para satisfacer sus necesidades. Esto ayudará a determinar los requerimientos del sistema de software y a desarrollar un sistema que satisfaga las necesidades del cliente.

### Respuesta Esperada:

El propósito de la identificación de actores en el proceso de especificación de requerimientos es determinar quiénes son las personas, sistemas o dispositivos que interactúan con el sistema y qué funcionalidades esperan de él.

### Contexto Relevante:

1. 13 Proceso para especificar requerimientos Actividades Obtención Requerimientos Identificación de actores Identificación Funcionalidades Personas Otro sistema Dispositivos Lo que se desea que el siste.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que el sistema .....

3. proceso para especificar requerimientos elicitation actividades obtencion requerimientos identificacion de actores identificacion funcionalidades personas otro sistema dispositivos lo que se desea que.....

### Métricas de Evaluación:

Answer Relevancy: 0.88 (Threshold: 0.5) - The score is 0.88 because the response is mostly on topic, but slightly deviates by discussing the mention of actor identification in documents rather than its purpose.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node highlighting the importance of 'identificacion de actores' and 'identificacion funcionalidades', the second node reiterating their significance, and the third node further emphasizing their role in specifying requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions 'identificacion de actores y funcionalidades', making it a seamless match.

Contextual Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the retrieval context is mostly relevant, with statements like 'identificacion de actores y funcionalidades' and 'identificacion de actores' directly addressing the input, but is brought down by irrelevant information like 'analisis de requerimientos' and 'validacion mediante pruebas...'.

## Caso de Prueba: 234 - Fallo

### Input del Usuario:

¿Cuál es el propósito de la técnica de "Lluvia de ideas" en el proceso de especificación del sistema?

### Respuesta Generada:

Según los documentos proporcionados, la técnica de "Lluvia de ideas" se menciona en el Documento 1 como una de las técnicas utilizadas en el proceso de especificación del sistema, específicamente en la etapa de determinación de requerimientos. El propósito de la técnica de "Lluvia de ideas" es recopilar y generar ideas y sugerencias de los usuarios, clientes y otros actores clave para identificar los requerimientos del sistema. Esta técnica se utiliza para fomentar la creatividad y la participación activa de los stakeholders en la definición de los requerimientos del sistema. En otras palabras, la técnica de "Lluvia de ideas" se utiliza para recopilar información y obtener requerimientos de los usuarios y otros interesados, lo que ayuda a identificar las necesidades y expectativas de los usuarios para el sistema de software.

### Respuesta Esperada:

El propósito de la técnica de "Lluvia de ideas" en el proceso de especificación del sistema es obtener requerimientos a través de talleres con los usuarios.

### Contexto Relevante:

1. 14 Proceso para especificar requerimientos Especificación del Sistema Determinación de Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudia.....

### Contexto Recuperado:

1. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....
2. presentacion de curso características del curso definiciones que es software que es ingeniería de software aplicar conceptos métodos y herramientas para crear aplicaciones de software diferencias escr.....
3. conceptos analisis es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema diseno es una descripcion o especificacion de proceso.....

### Métricas de Evaluación:

Answer Relevancy: 0.57 (Threshold: 0.5) - The score is 0.57 because the answer partially addresses the input question by mentioning the technique's usage, but lacks explicit explanations of its purpose, resulting in irrelevant statements.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node being directly related to the purpose of 'Lluvia de ideas' technique, and the lower-ranked nodes, like node 2 and node 3, are not directly related to the purpose of 'Lluvia de ideas' technique in the context of system specification, as they are about 'presentacion de curso', 'conceptos metodos y herramientas', 'conceptos analisis', and 'diseno es una descripcion'.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless connection between the technique and the system specification process.

Contextual Relevancy: 0.15 (Threshold: 0.5) - The score is 0.15 because the retrieval context is mostly irrelevant to the input, with reasons stating that the statements are too broad, vague, or unrelated to the purpose of the 'Lluvia de ideas' technique, as seen in quotes such as 'The statement is too broad and does not specifically relate to the purpose of the 'Lluvia de ideas' technique.' and 'The statement is not related to the purpose of the 'Lluvia de ideas' technique.'

## Caso de Prueba: 235 - Fallo

### Input del Usuario:

¿Cuál es el propósito de realizar entrevistas con los usuarios en el proceso de determinación de requerimientos?

### Respuesta Generada:

Según el Documento 5, el propósito de realizar entrevistas con los usuarios en el proceso de determinación de requerimientos es identificar el propósito de la entrevista, mediante preguntas que permitan entender con claridad cada cosa que el cliente desea. Las entrevistas con los usuarios y las partes interesadas son importantes para crear un software adecuado. Además, las entrevistas permiten familiarizarse con el vocabulario del negocio y tomar apuntes o grabar la reunión. En resumen, el propósito de las entrevistas es recopilar y descubrir los requerimientos de un sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave, para determinar las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software.

### Respuesta Esperada:

El propósito de realizar entrevistas con los usuarios en el proceso de determinación de requerimientos es obtener información valiosa y precisa sobre los requerimientos del sistema a través de la interacción directa con los usuarios finales.

### Contexto Relevante:

1. 14 Proceso para especificar requerimientos Especificación del Sistema Determinación de Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudia.....

### Contexto Recuperado:

1. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....
2. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....
3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the response partially addresses the purpose of conducting user interviews, but is brought down by circular and logistical statements

that stray from the main point.

Contextual Precision: 0.58 (Threshold: 0.5) - The score is 0.58 because the first node, which is irrelevant, is ranked higher than the second and third nodes, which are relevant, due to reasons such as 'entender claramente...definir adecuadamente los requerimientos' not directly addressing the purpose of conducting user interviews, whereas 'entrevistas con los usuarios' is mentioned as a technique for specifying requirements and 'elicitation es el proceso de recopilar y descubrir los requerimientos...a traves de la comunicacion directa con los interesados como usuarios finales' clearly explains the purpose.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire sentence in the expected output is perfectly attributed to nodes 2 and 3 in the retrieval context, providing a precise match.

Contextual Relevancy: 0.25 (Threshold: 0.5) - The score is 0.25 because the retrieval context is mostly irrelevant, with statements like 'documentacion', 'talleres lluvia de ideas', and 'prototipos' not being related to the purpose of conducting user interviews in the requirement determination process, but it does contain some relevant information like 'definir adecuadamente los requerimientos' and 'entrevistas con los usuarios'.



## Caso de Prueba: 236 - Éxito

### Input del Usuario:

¿Cuál es el proceso para especificar requerimientos en la creación de un SRS?

### Respuesta Generada:

Según los documentos proporcionados, el proceso para especificar requerimientos en la creación de un SRS (Software Requirement Specification) consta de las siguientes etapas: 1. Elicitación: Es el proceso de recopilar y descubrir los requerimientos del sistema de software a través de la comunicación directa con los interesados, como usuarios finales, clientes y otros actores clave. Esto implica la identificación de actores y funcionalidades. 2. Análisis de requerimientos: Es el proceso de identificar y entender las necesidades y expectativas de los usuarios y otras partes interesadas para un sistema de software. 3. Especificación: Consiste en definir qué funciones y características debe tener el software, así como sus restricciones y condiciones de operación. 4. Documentación: Se crea un documento de especificación de requerimientos que contiene la información recopilada y analizada. 5. Validación: Se realizan pruebas para asegurar que los requerimientos fueron entendidos y que el producto final cumpla con los objetivos del proyecto y satisfaga las necesidades del cliente. En resumen, el proceso para especificar requerimientos en la creación de un SRS implica la elicitación de los requerimientos, su análisis, especificación, documentación y validación.

### Respuesta Esperada:

El proceso para especificar requerimientos en la creación de un SRS (Software Requirement Specification) es el siguiente: 1. Determinación de requerimientos: Se obtienen los requerimientos a través de técnicas como entrevistas con los usuarios, talleres (lluvia de ideas), estudio de sistemas parecidos, encuestas y prototipos. 2. Documentación: Se documentan los requerimientos obtenidos. 3. Validación: Se comprueban las características de los requerimientos para asegurarse de que sean claros, completos, correctos, consistentes y no ambiguos. 4. Especificación del sistema: Se crea el SRS con los requerimientos especificados y validados.

### Contexto Relevante:

1. 14 Proceso para especificar requerimientos Especificación del Sistema Determinación de Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudia.....

### Contexto Recuperado:

1. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

2. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

3. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar

siste.....

### **Métricas de Evaluación:**

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a clear and concise process for specifying requirements in creating an SRS!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are relevant to the input, with the first node explaining the process of elicitation, analysis, specification, and validation, the second node listing techniques for obtaining requirements, and the third node reiterating the importance of determining and documenting requirements, making them all perfectly ranked.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because all sentences in the expected output are perfectly attributed to nodes in the retrieval context, with sentences 1 and 4 matching the 1st node and sentences 2 and 3 matching the 2nd node, resulting in a flawless contextual recall.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context provides a comprehensive and detailed explanation of the process for specifying requirements in creating an SRS, perfectly aligning with the input question.

## Caso de Prueba: 237 - Éxito

### Input del Usuario:

¿Cuál es el propósito de crear prototipos en el proceso de determinar los requerimientos del sistema?

### Respuesta Generada:

Según el Documento 5, el propósito de crear prototipos en el proceso de determinar los requerimientos del sistema es para la validación de los requisitos. Específicamente, se menciona que el prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario, y que existen dos tipos principales de prototipos de interfaz de usuario: desechables y evolutivos. Los prototipos se utilizan para presentar una versión aproximada del sistema a los usuarios y otros interesados, con el fin de recopilar feedback y asegurarse de que los requisitos se están entendiendo correctamente. De esta manera, se puede validar que los requisitos sean correctos antes de proceder con el desarrollo del sistema. En resumen, el propósito de crear prototipos es para la validación y clarificación de los requisitos del sistema, a fin de asegurarse de que se cumplan los objetivos del proyecto y se satisfagan las necesidades del cliente.

### Respuesta Esperada:

El propósito de crear prototipos en el proceso de determinar los requerimientos del sistema es comprobar las características de los requerimientos.

### Contexto Relevante:

1. 14 Proceso para especificar requerimientos Especificación del Sistema Determinación de Requerimientos Documentación Obtención Validación Entrevistas con los usuarios Talleres (Lluvia de ideas) Estudia.....

### Contexto Recuperado:

1. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....

2. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

3. entender claramente cual es el objetivo de un proyecto realizar la estimacion de costos y tiempo de desarrollo crear un plan de trabajo establecer prioridades definir adecuadamente los requerimientos .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question, providing a clear and relevant response!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which has a reason of 'entender claramente cual es el objetivo de un proyecto...' that is not directly related to creating prototypes for determining system requirements.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire sentence in the expected output is perfectly attributed to the 1st and 2nd nodes in the retrieval context, showcasing a seamless alignment!

Contextual Relevancy: 0.56 (Threshold: 0.5) - The score is 0.56 because the retrieval context has many irrelevant statements, such as 'documentacion', 'entrevistas con los usuarios', and 'estudiar sistemas parecidos', which are not directly related to the purpose of creating prototypes in the process of determining system requirements, diluting the relevance of the relevant statements.

## Caso de Prueba: 238 - Éxito

### Input del Usuario:

¿Cuál es el propósito de la validación de expertos en la técnica de validación de requerimientos?

### Respuesta Generada:

Según el Documento 1 y el Documento 7, el propósito de la validación de expertos en la técnica de validación de requerimientos es que personas con experiencia revisen los requerimientos y aprueben o rechacen el requerimiento. En otras palabras, la validación de expertos tiene como objetivo asegurar que los requisitos del sistema sean precisos y correctos mediante la revisión y aprobación de expertos en el campo.

### Respuesta Esperada:

El propósito de la validación de expertos en la técnica de validación de requerimientos es que personas con experiencia revisen los requerimientos y aprueben o rechazan el requerimiento.

### Contexto Relevante:

1. 15 Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prot.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....
3. proceso para especificar requerimientos sirve para determinar los requerimientos elicitation es el proceso de recopilar y descubrir los requerimientos de un sistema de software a traves de la comunic.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, like node 1, which clearly addresses the question, are ranked higher than irrelevant nodes, like node 2 and node 3, which do not mention the purpose of expert validation in requirement validation technique.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which explicitly mentions experts reviewing and approving/rejecting requirements.

Contextual Relevancy: 0.81 (Threshold: 0.5) - The score is 0.81 because the retrieval context contained some irrelevant information like 'prototipado de interfaz de usuario' and 'recorrido de bpm', but also had relevant statements like 'validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento' that directly relate to the purpose of expert validation in requirement validation technique.

## Caso de Prueba: 239 - Éxito

### Input del Usuario:

¿Cuál es el propósito principal de la técnica de prototipado de interfaz de usuario en la validación de requerimientos?

### Respuesta Generada:

Según el Documento 1, el propósito principal de la técnica de prototipado de interfaz de usuario en la validación de requerimientos es la representación aproximada de la interfaz de usuario, lo que permite la validación de los requisitos mediante la creación de un modelo de pantallas o interfaz de usuario que represente el flujo visual de una aplicación de software. En otras palabras, el prototipado de interfaz de usuario tiene como objetivo principal permitir la validación de los requisitos funcionales y no funcionales de una aplicación, mediante la creación de una versión preliminar de la interfaz de usuario que pueda ser revisada y aprobada o rechazada por los expertos y los stakeholders involucrados en el proyecto. Esta técnica permite identificar y corregir problemas de diseño y funcionalidad en una etapa temprana del desarrollo, lo que reduce el riesgo de errores y costos adicionales en el futuro.

### Respuesta Esperada:

El propósito principal de la técnica de prototipado de interfaz de usuario en la validación de requerimientos es la representación aproximada de la interfaz de usuario para validar los requisitos.

### Contexto Relevante:

1. 15 Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prot.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. prototipado de interfaz de usuario modelo de pantallas mockup user interfaces fronend ui...
3. interfaz de usuario logica de la aplicacion interfaz de datos api frontend backend arquitectura para aplicaciones desacopladas basadas en api bdatos web service servidor web tipos de estilos architect.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a direct and accurate answer to the question!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 which directly addresses the question and node 2 which mentions prototipado de interfaz de usuario in the context of requirement validation, are ranked higher than the irrelevant node 3, which is not directly related to the purpose of prototyping in requirement validation.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly matched with the 2nd node in retrieval context, which precisely describes the purpose of user interface prototyping.

Contextual Relevancy: 0.73 (Threshold: 0.5) - The score is 0.73 because the retrieval context is mostly irrelevant, with statements about 'validacion de expertos', 'recorrido de bpm', and 'arquitectura para aplicaciones desacopladas' that do not relate to the purpose of prototipado de interfaz de usuario, but it does contain some relevant statements about user interface prototyping, such as 'prototipado de interfaz de usuario es una tecnica de representacion aproximada de la interfaz de usuario'.



## Caso de Prueba: 240 - Fallo

### Input del Usuario:

¿Cuál es el propósito principal de los prototipos desechables en la validación de requerimientos?

### Respuesta Generada:

Según los documentos 1 y 9, el propósito principal de los prototipos desechables en la validación de requerimientos es la representación aproximada de la interfaz de usuario para la validación de los requisitos. Estos prototipos se utilizan solo para la validación de los requisitos y posteriormente se desechan. De esta forma, los prototipos desechables permiten a los usuarios y expertos revisar y aprobar o rechazar los requisitos del sistema, sin tener que invertir tiempo y recursos en una implementación completa.

### Respuesta Esperada:

El propósito principal de los prototipos desechables en la validación de requerimientos es la validación de los requisitos.

### Contexto Relevante:

1. 15 Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prot.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. proceso para especificar requerimientos especificacion del sistema determinar los requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar siste.....
3. proceso para especificar requerimientos especificacion del sistema determinacion de requerimientos documentacion obtencion validacion entrevistas con los usuarios talleres lluvia de ideas estudiar sis.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant information!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are correctly ranked, with the first node, which directly answers the question about the main purpose of disposable prototypes in requirement validation, ranked highest.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 2nd node in the retrieval context, which explicitly mentions 'prototipos desechables... se utilizan solo para la validacion de los requisitos'.

Contextual Relevancy: 0.12 (Threshold: 0.5) - The score is 0.12 because the retrieval context mostly contains irrelevant statements that do not specifically mention or relate to the purpose of disposable prototypes in requirement validation, as stated in the reasons for irrelevancy, such as 'The statement does not specifically relate to the purpose of disposable prototypes in requirement validation.'

## Caso de Prueba: 241 - Fallo

### Input del Usuario:

¿Cuál es la principal diferencia entre un prototipo evolutivo y uno desechable en el contexto de la validación de requerimientos?

### Respuesta Generada:

De acuerdo con el Documento 1, la principal diferencia entre un prototipo evolutivo y uno desechable en el contexto de la validación de requerimientos es que: \* Un prototipo desechable se utiliza solo para la validación de los requisitos y posteriormente se desecha. \* Un prototipo evolutivo, una vez utilizado para la validación de los requisitos, se mejora su calidad y se convierte progresivamente en el producto final. En otras palabras, los prototipos desechables son descartados después de que se han cumplido sus objetivos de validación, mientras que los prototipos evolutivos se refinan y mejoran para convertirse en la versión final del producto.

### Respuesta Esperada:

La principal diferencia entre un prototipo evolutivo y uno desechable es que el prototipo evolutivo se mejora y se convierte progresivamente en el producto final después de ser utilizado para la validación de los requisitos, mientras que el prototipo desechable se utiliza solo para la validación de los requisitos y posteriormente se desecha.

### Contexto Relevante:

1. 15 Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prot.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....
3. tareas investigar la diferencia entre get y post get 1 funcion el metodo get se utiliza para solicitar datos de un servidor es decir se usa para recuperar informacion por ejemplo una pagina web o los .....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the actual output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all irrelevant nodes, such as the 2nd and 3rd nodes about 'tareas investigar la diferencia entre get y post', are correctly ranked lower

than the 1st node, which is directly related to the question about prototypes in the context of requirement validation.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly explains the difference between evolutionary and disposable prototypes.

Contextual Relevancy: 0.11 (Threshold: 0.5) - The score is 0.11 because the retrieval context is mostly irrelevant to the input, with many statements discussing HTTP requests and other unrelated topics, as stated in the reasons for irrelevancy, such as 'requerimientos no funcionales tecnicas', 'validacion de expertos', and 'recorrido de bpm', whereas only two statements are relevant, mentioning 'prototipado de interfaz de usuario' and 'los dos tipos principales de prototipos de interfaz de usuario son desechables y evolutivos'.

## Caso de Prueba: 242 - Fallo

### Input del Usuario:

¿Cuál es el algoritmo de alto nivel que se utiliza en el recorrido de BPM para validarlos?

### Respuesta Generada:

Según el Documento 1, el recorrido de BPM (Business Process Model) se utiliza para mostrar todo el proceso, y se utiliza un algoritmo de alto nivel para validar los requerimientos. Sin embargo, no se especifica explícitamente cuál es el algoritmo de alto nivel utilizado. Es importante destacar que los demás documentos no proporcionan información adicional sobre este tema específico. Por lo tanto, no se puede determinar con certeza cuál es el algoritmo de alto nivel utilizado en el recorrido de BPM para validar los requerimientos.

### Respuesta Esperada:

El algoritmo de alto nivel que se utiliza en el recorrido de BPM para validarlos es un BPM que muestre todo el proceso.

### Contexto Relevante:

1. 15 Requerimientos no funcionales Técnicas para validación de requerimientos 1. Validación de expertos Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento
2. Prot.....

### Contexto Recuperado:

1. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
2. requerimientos no funcionales tecnicas para validacion de requerimientos 1 validacion de expertos personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento 2 prototipa.....
3. introduccion a bpmn business process model and notation desafios de esta tecnica decidir el nivel de detalle en que se modelara el proceso de negocio como manejar las variaciones en el proceso de nego.....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the actual output partially addresses the input question, but includes some irrelevant information about the purpose of the BPM walkthrough and the lack of information in other documents.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all the relevant nodes, such as node 1 and node 2, are correctly ranked higher than the irrelevant node 3, which talks about 'introduccion a bpmn business process model and notation desafios de esta tecnica...' and is not

directly related to the question.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 3rd node in the retrieval context, which matches the sentence exactly.

Contextual Relevancy: 0.13 (Threshold: 0.5) - The score is 0.13 because the retrieval context is mostly irrelevant, with statements discussing expert validation, user interface prototyping, non-functional requirements, and other unrelated topics, as stated in the reasons for irrelevancy, such as 'The statement is not related to the algorithm used in BPM traversal for validation.' and 'The statement is about user interface prototyping, not the algorithm used in BPM traversal.'

## Caso de Prueba: 243 - Éxito

### Input del Usuario:

Cuál es el propósito principal de la ceremonia de retrospectiva en Scrum?

### Respuesta Generada:

Según el Documento 1, el propósito principal de la ceremonia de retrospectiva en Scrum es analizar cómo les fue en el último sprint, es decir, cómo trabajaron, qué problemas tuvieron, qué cosas funcionaron bien y qué no. El objetivo es que el equipo de desarrollo y el Scrum Master analicen cómo ha sido su manera de trabajar durante el sprint, si están consiguiendo los objetivos a los que se comprometieron al inicio del sprint, y si el incremento de producto que acaba de demostrar al cliente era lo que él esperaba o no.

### Respuesta Esperada:

El propósito principal de la ceremonia de retrospectiva en Scrum es reflexionar sobre el proceso de trabajo y mejorar la eficiencia del equipo.

### Contexto Relevante:

1. 18 Metodologías ágiles...

### Contexto Recuperado:

1. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....
2. practicas agiles en scrum sprint planning meeting el sprint planning en scrum es una reunion al inicio de cada sprint donde el product owner el scrum team y el scrum master colaboran para definir que .....
3. practicas agiles en scrum retrospectiva el objetivo es analizar como les fue en el ultimo sprint como trabajamos que problemas tuvimos que cosas funcionaron bien y cuales no participan el equipo de de.....

### Métricas de Evaluación:

Answer Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the response correctly addresses the input question, but includes some irrelevant statements that detract from its overall relevance.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node 1 in the retrieval contexts directly addresses the question, and node 3, which also answers the question, is ranked higher than node 2, which is not relevant to the topic of the retrospective ceremony in Scrum, as it talks about sprint planning meeting.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to nodes 1 and 3 in the retrieval context, showcasing a flawless alignment.

Contextual Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the retrieval context perfectly aligns with the input, providing a clear and comprehensive explanation of the purpose of the retrospective ceremony in Scrum, as stated in the relevant statements, such as 'el objetivo es analizar como les fue en el ultimo sprint' and 'el proposito es alinear al equipo en torno a un plan realista y alcanzable maximizando el valor del producto'.



## Caso de Prueba: 244 - Éxito

### Input del Usuario:

¿Cuál es el enfoque principal del desarrollo de software ágil?

### Respuesta Generada:

Según los documentos proporcionados, el enfoque principal del desarrollo de software ágil es iterativo, enfatizando la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo. En otras palabras, se centra en entregar pequeñas partes del software de forma continua y colaborativa, en lugar de intentar entregar todo de una vez al final del proyecto. Esto se refleja en documentos como el 1, 2 y 3, que definen el desarrollo de software ágil como un enfoque que se centra en la entrega incremental y la colaboración en equipo. Además, el documento 6 destaca que "el software funcionando es la medida principal de progreso" y que "los procesos ágiles promueven el desarrollo sostenible", lo que sugiere que el enfoque principal es la entrega continua de software funcional.

### Respuesta Esperada:

El enfoque principal del desarrollo de software ágil es enfatizar la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo.

### Contexto Relevante:

1. 19 Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el ap.....

### Contexto Recuperado:

1. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly aligns with the input, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all nodes in the retrieval contexts are perfectly ranked, with the top 2 nodes directly addressing the question and providing the main focus of agile software development, while the 3rd node, which provides general information, is correctly ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly matched with the 1st node in the retrieval context, which conveys the same idea of emphasizing incremental delivery, team collaboration, continuous planning, and continuous learning.

Contextual Relevancy: 0.60 (Threshold: 0.5) - The score is 0.60 because the retrieval context has some relevant statements that describe agile software development, such as 'agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software' and 'enfatan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendizaje continuo', but is also cluttered with irrelevant information, like 'scrumcolombia desarrollo de software i' and phrases that don't directly relate to the main focus of agile software development.

## Caso de Prueba: 245 - Éxito

### Input del Usuario:

¿Cuál es el enfoque principal del desarrollo de software ágil en lugar de intentar entregarlo todo de una vez cerca del final?

### Respuesta Generada:

Según el Documento 1 y Documento 2, el enfoque principal del desarrollo de software ágil es entregar de manera incremental, enfatizando la colaboración en equipo, la planificación continua y el aprendizaje continuo. Esto se opone a intentar entregar todo al final del proyecto. Este enfoque iterativo permite entregar pequeñas partes del software de manera regular, lo que facilita la retroalimentación, la adaptación a cambios y la entrega de un producto de alta calidad.

### Respuesta Esperada:

El enfoque principal del desarrollo de software ágil es la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo.

### Contexto Relevante:

1. 19 Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el ap.....

### Contexto Recuperado:

1. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. desarrollo de software ágil el desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles se concibieron como una alternativa a las pract.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question without any irrelevant statements!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which clearly address the question, are ranked higher than irrelevant nodes, like node 3, which only provides general information about agile software development.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, which clearly describes the agile software development approach.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context contained some relevant statements like 'en lugar de intentar entregarlo todo de una vez cerca del final desarrollo de software agil' and 'enfatan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendizaje continuo', which align with the input's focus on agile software development, but was brought down by irrelevant information like 'scrumcolombia desarrollo de software i' and a summary of agile software development.

## Caso de Prueba: 246 - Éxito

### Input del Usuario:

¿Cuál es el enfoque principal de la entrega incremental en el desarrollo de software ágil?

### Respuesta Generada:

Según los documentos proporcionados, el enfoque principal de la entrega incremental en el desarrollo de software ágil es entregar pequeñas partes del software funcionando de manera continua, en lugar de intentar entregarlo todo de una vez al final del proyecto. Esto se enfatiza en los Documentos 1, 2, 5, 6 y 7, que mencionan la entrega incremental y pequeñas como una característica fundamental del desarrollo de software ágil. En particular, el Documento 5 y 6 mencionan que la metodología ágil Scrum se basa en la entrega de pequeños incrementos de software, lo que permite una retroalimentación continua y ajustes en el proceso de desarrollo. Esto se logra a través de la planificación continua, la colaboración en equipo y el aprendizaje continuo. En resumen, el enfoque principal de la entrega incremental en el desarrollo de software ágil es entregar pequeñas partes del software funcionando de manera continua, lo que permite una mayor flexibilidad, adaptabilidad y entrega de valor al cliente de manera temprana y sostenida.

### Respuesta Esperada:

El enfoque principal de la entrega incremental en el desarrollo de software ágil es enfatizar la entrega de pequeñas partes del software en lugar de intentar entregarlo todo de una vez cerca del final.

### Contexto Relevante:

1. 19 Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el ap.....

### Contexto Recuperado:

1. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### Métricas de Evaluación:

Answer Relevancy: 0.90 (Threshold: 0.5) - The score is 0.90 because the response is mostly relevant, but slightly digressed to emphasize documents, which is not the main focus of incremental delivery.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, are ranked higher than the irrelevant node 3, which only provides general information

about agile software development without focusing on incremental delivery.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the entire expected output is perfectly attributed to the 1st node in the retrieval context, showcasing a seamless alignment.

Contextual Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because although the retrieval context contained irrelevant information like 'desarrollo de software agil resumen' and 'scrumcolombia desarrollo de software i', it also had relevant statements like 'agil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental' that directly relate to the main focus of incremental delivery in agile software development.

## Caso de Prueba: 247 - Fallo

### Input del Usuario:

¿Cuál es el beneficio principal de la colaboración en equipo en el desarrollo de software ágil?

### Respuesta Generada:

Según el Documento 3, "las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados". Esto sugiere que la colaboración en equipo en el desarrollo de software ágil permite que los miembros del equipo compartan sus ideas y perspectivas, lo que a su vez conduce a la creación de soluciones más efectivas y sostenibles. Además, la colaboración en equipo en el desarrollo de software ágil fomenta la comunicación cara a cara, presencial o virtual, que es considerada el método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros (Documento 7). En resumen, el beneficio principal de la colaboración en equipo en el desarrollo de software ágil es que permite la emergencia de soluciones más efectivas y sostenibles, gracias a la compartición de ideas y perspectivas entre los miembros del equipo.

### Respuesta Esperada:

El beneficio principal de la colaboración en equipo en el desarrollo de software ágil es la entrega incremental y continua del producto.

### Contexto Relevante:

1. 19 Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el ap.....

### Contexto Recuperado:

1. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
2. ágil es un termino que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental la colaboracion en equipo la planificacion continua y el aprendi.....
3. 07 el software funcionando es la medida principal de progreso los procesos agiles promueven el desarrollo sostenible la atencion continua a la excelencia tecnica y al buen diseno mejora la agilidad la.....

### Métricas de Evaluación:

Answer Relevancy: 0.83 (Threshold: 0.5) - The score is 0.83 because the response is mostly relevant to the input, but slightly diverges to discuss communication methods, which is not the main focus of the question.

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which directly address the question by mentioning the importance of collaboration in agile software development, are ranked higher than the irrelevant node, node 3, which focuses on the

importance of working software and team reflection.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the 1st node in retrieval context, which explicitly mentions key phrases like 'la entrega incremental' and 'la colaboracion en equipo'.

Contextual Relevancy: 0.40 (Threshold: 0.5) - The score is 0.40 because the retrieval context mostly mentioned irrelevant concepts like 'la planificacion continua', 'el aprendizaje continuo', and 'entregarlo todo de una vez al final del proyecto', which are not directly related to the main benefit of team collaboration in agile software development, while only a few statements like 'la colaboracion en equipo' and 'el desarrollo de software agil' were relevant.



## Caso de Prueba: 248 - Fallo

### Input del Usuario:

¿Cuáles son los 4 valores fundamentales del desarrollo de software ágil?

### Respuesta Generada:

Según los documentos proporcionados, especialmente el Documento 2, los 4 valores fundamentales del desarrollo de software ágil son: 1. Individuos e interacciones 2. Software funcionando 3. Colaboración con el cliente 4. Respuesta al cambio Estos valores son esenciales para understanding el enfoque ágil en el desarrollo de software y guían la forma en que los equipos de desarrollo trabajan juntos para entregar software de alta calidad.

### Respuesta Esperada:

Según el contexto, los 4 valores fundamentales del desarrollo de software ágil no se mencionan explícitamente. Sin embargo, se hace referencia a que están "Descrita por" y "Definida por", lo que sugiere que se están refiriendo a los 4 valores del Manifiesto Ágil, que son: \* Individuals and interactions (Individuos y interacciones) \* Working software (Software funcionando) \* Customer collaboration (Colaboración con el cliente) \* Responding to change (Responder al cambio) Sin embargo, es importante destacar que estos valores no se mencionan explícitamente en el contexto proporcionado.

### Contexto Relevante:

1. 20 Desarrollo de software ágil El desarrollo de software ágil representa una forma de pensar y hacer las cosas. Ágil es una Mentalidad Ser Ágil 4 Valores Descrita por Definida por 12 Principios Manife.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo de software agil representa una forma de pensar y hacer las cosas agil es una mentalidad ser agil 4 valores descrita por definida por 12 principios manifestad.....
2. desarrollo de software agil agil es una forma de pensar y de hacer las cosas y un comportamiento durante un proceso de desarrollo agil es una mentalidad individuos e interacciones software funcionando.....
3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which explicitly mention the 4 values of the Agile Manifesto, are ranked higher than the

irrelevant node 3, which only provides general information about agile development.

Contextual Recall: 0.60 (Threshold: 0.5) - The score is 0.60 because the output correctly attributes the 4 values of the Agile Manifesto to nodes in the retrieval context, specifically nodes 1 and 2, but lacks explicit mentions in the context.

Contextual Relevancy: 0.22 (Threshold: 0.5) - The score is 0.22 because the retrieval context mostly contains irrelevant information, such as repetitions of the letter 'p', and only a few statements like 'agil es una mentalidad individuos e interacciones software funcionando colaboracion con el cliente respuesta al cambio' and 'descrita por 4 valores definido por 12 principios' that vaguely relate to the input.

## Caso de Prueba: 249 - Éxito

### Input del Usuario:

¿Cuáles son los 4 valores que describen el desarrollo de software ágil?

### Respuesta Generada:

Según los Documentos 1 y 2, los 4 valores que describen el desarrollo de software ágil son: 1. Individuos e interacciones 2. Software funcionando 3. Colaboración con el cliente 4. Respuesta al cambio Estos 4 valores son fundamentales en la mentalidad y enfoque del desarrollo de software ágil, y guían las prácticas y principios que se aplican en este enfoque.

### Respuesta Esperada:

Según el contexto, los 4 valores que describen el desarrollo de software ágil no se mencionan explícitamente. Sin embargo, se habla de que el desarrollo de software ágil se describe a través de 4 valores y 12 principios, pero no se proporciona la lista de estos valores.

### Contexto Relevante:

1. 20 Desarrollo de software ágil El desarrollo de software ágil representa una forma de pensar y hacer las cosas. Ágil es una Mentalidad Ser Ágil 4 Valores Descrita por Definida por 12 Principios Manife.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo de software agil representa una forma de pensar y hacer las cosas agil es una mentalidad ser agil 4 valores descrita por definida por 12 principios manifestad.....
2. desarrollo de software agil agil es una forma de pensar y de hacer las cosas y un comportamiento durante un proceso de desarrollo agil es una mentalidad individuos e interacciones software funcionando.....
3. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....

### Métricas de Evaluación:

Answer Relevancy: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly addresses the input question, providing a precise and relevant answer!

Contextual Precision: 1.00 (Threshold: 0.5) - The score is 1.00 because all relevant nodes, such as node 1 and node 2, which provide the 4 values describing agile software development, are ranked higher than the irrelevant node 3, which only talks about the methodology and principles without providing the required list.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output accurately captures the essence of the retrieval context, specifically the 1st node's mention of '4 valores descrita por definida

por 12 principios', demonstrating a perfect alignment.

Contextual Relevancy: 0.50 (Threshold: 0.5) - The score is 0.50 because the retrieval context is only partially relevant, with statements like 'descrita por 4 valores definido por 12 principios' and 'agil es una mentalidad individuos e interacciones software funcionando colaboracion con el cliente respuesta al cambio' providing some connection to the 4 values of agile software development, but being overshadowed by irrelevant information.

## Caso de Prueba: 250 - Éxito

### Input del Usuario:

¿Cuántos valores y principios se describen en el desarrollo de software ágil?

### Respuesta Generada:

Según los documentos proporcionados, el desarrollo de software ágil se describe como una mentalidad que tiene 4 valores y 12 principios. Esto se menciona en los documentos 1, 3 y otros. Por lo tanto, la respuesta a la pregunta es: 4 valores y 12 principios.

### Respuesta Esperada:

Según el contexto, se describen 4 valores y 12 principios en el desarrollo de software ágil.

### Contexto Relevante:

1. 20 Desarrollo de software ágil El desarrollo de software ágil representa una forma de pensar y hacer las cosas. Ágil es una Mentalidad Ser Ágil 4 Valores Descrita por Definida por 12 Principios Manife.....

### Contexto Recuperado:

1. desarrollo de software agil el desarrollo de software agil representa una forma de pensar y hacer las cosas agil es una mentalidad ser agil 4 valores descrita por definida por 12 principios manifestad.....
2. desarrollo de software agil el desarrollo agil de software es un paradigma usado en las metodologias de desarrollo de software basado en procesos agiles se concibieron como una alternativa a las pract.....
3. desarrollo de software agil agil es una forma de pensar y de hacer las cosas y un comportamiento durante un proceso de desarrollo agil es una mentalidad individuos e interacciones software funcionando.....

### Métricas de Evaluación:

Answer Relevancy: 0.75 (Threshold: 0.5) - The score is 0.75 because the response is mostly relevant, but was slightly sidetracked by mentioning documents, which isn't directly related to the number of values and principles in agile software development.

Contextual Precision: 0.83 (Threshold: 0.5) - The score is 0.83 because the top-ranked node directly answers the question, and the third node reiterates the correct information, but the second node, ranked 2, is an irrelevant node that only explains the concept of agile software development, which should be ranked lower.

Contextual Recall: 1.00 (Threshold: 0.5) - The score is 1.00 because the output perfectly matches the retrieval context, with the sentence accurately describing 4 values and 12 principles, as stated in the 1st node.

Contextual Relevancy: 0.80 (Threshold: 0.5) - The score is 0.80 because the retrieval context contained some irrelevant information, such as repetitive 'p's and specific organization details, but also provided relevant statements like '4 valores' and 'descrita por 12 principios' that directly relate to the values and principles of agile software development.