17-11-2024

Ángel Roldán Rabanal

1 DAM Digitalización 24/25

Exploración del Lenguaje Natural y el Contexto en Herramientas de IA Generativa

Índice

[Parte 1: Ejemplos guiados, comparación y uso de contexto 2](#_Toc182928686)

[1. Pruebas básicas 2](#_Toc182928687)

[2. Comparación: 2](#_Toc182928688)

[3. Uso del contexto y roles: 3](#_Toc182928689)

[4. Pruebas con diferentes idiomas: 4](#_Toc182928690)

[5. Reformulación de consultas: 5](#_Toc182928691)

# Parte 1: Ejemplos guiados, comparación y uso de contexto

## 1. Pruebas básicas

**Consulta 1:** “¿Cuáles son los principales motores gráficos usados en el desarrollo de videojuegos?”

**Respuesta:** Los principales motores gráficos son Unreal Engine, Unity, y Godot. Cada uno ofrece herramientas para diseño 3D, scripting y físicas avanzadas.

**Consulta 2:** “Motores video juegos mejores, ¿qué son?”

**Respuesta:** Unity, Unreal. Bueno para juegos.

**Consulta 3:** “Háblame de videojuegos.”

**Respuesta:** Los videojuegos son programas interactivos diseñados para entretener, enseñar o contar historias. Pueden desarrollarse para múltiples plataformas como consolas, PC o móviles.

**Consulta 4:** “Cómo influye la elección del motor gráfico en el desarrollo de un videojuego, tanto en términos de calidad como de tiempo de desarrollo?”

**Respuesta:** La elección del motor gráfico influye directamente en la calidad visual y el tiempo necesario para el desarrollo. Motores como Unreal Engine ofrecen gráficos avanzados, ideales para juegos AAA, pero pueden requerir más tiempo para optimización. Por otro lado, motores como Unity o Godot son más ligeros y eficientes para desarrollos rápidos o proyectos indie. Factores como la experiencia del equipo y el tipo de juego también son determinantes en esta decisión.

## 2. Comparación:

**Respuesta más completa:**

La consulta más completa fue la número 4 porque aborda múltiples variedades de temas relacionadas con el desarrollo, como calidad, tiempo y costos. Esto refleja la importancia de formular preguntas claras y específicas.

**Errores gramaticales o ambigüedad:**

La consulta 2, con errores gramaticales, generó respuestas más simples y menos útiles. Esto demuestra que la herramienta tiene dificultades para procesar consultas mal formuladas, lo que afecta la precisión y detalle de las respuestas.

**Limitaciones observadas:**

La herramienta tiende a asumir información implícita, lo que puede resultar en respuestas incompletas si no se especifican todos los detalles relevantes en la consulta.

Las consultas ambiguas o muy generales no permiten explorar el potencial completo de la herramienta.

## 3. Uso del contexto y roles:

**Consulta sin contexto:**

"¿Qué beneficios ofrece Unity en el desarrollo de videojuegos?"

**Respuesta:** Unity es un motor gráfico usado para crear juegos 2D y 3D. Ofrece herramientas multiplataforma y una gran comunidad de soporte.

**Consulta con contexto:**

"Soy un estudiante de desarrollo de aplicaciones que quiere aprender Unity para crear un juego en 2D. ¿Qué beneficios específicos me ofrece?"

**Respuesta**: Unity es una excelente opción para juegos en 2D. Te permite crear animaciones simples y mecánicas interactivas con C#. Además, cuenta con plantillas para empezar rápido y una amplia biblioteca de recursos que puedes reutilizar, ideal para proyectos.

**Observaciones:**

La consulta con contexto genera una respuesta más personalizada y específica al incluir detalles sobre el propósito del usuario (crear un juego).

Sin contexto, la respuesta es más genérica, enfocada solo en las características generales de Unity.

**Consulta con un rol definido:**

**Rol 1:** Actúa como un desarrollador indie experimentado.

"Como desarrollador indie, ¿qué consejos me darías para usar Unity en mi primer videojuego?"

**Respuesta:** Si estás comenzando con Unity, enfócate primero en un proyecto pequeño y manejable. Usa los tutoriales oficiales y aprovecha los assets gratuitos de la Unity Asset Store para acelerar tu desarrollo. Además, aprende a usar el sistema de prefabs, que te ahorrará tiempo en la creación de elementos reutilizables.

**Rol 2:** Actúa como un profesor explicando Unity a principiantes.

"¿Cómo le explicarías Unity a un grupo de estudiantes que no tienen experiencia en desarrollo?"

**Respuesta:** Unity es como una caja de herramientas para crear videojuegos. Puedes diseñar niveles, personajes y mecánicas usando su interfaz. Es fácil de usar para principiantes porque puedes arrastrar y soltar elementos, y su lenguaje de programación, C#, es uno de los más populares y accesibles.

**Observaciones:**

Con rol de desarrollador indie: La respuesta se enfoca en consejos prácticos, adaptados a las necesidades de un creador individual.

Con rol de profesor: La explicación es más sencilla, con un tono didáctico, ideal para principiantes.

## 4. Pruebas con diferentes idiomas:

**Consulta en inglés sin contexto:**

"What benefits does Unity offer for game development?"

**Respuesta**: Unity is a game engine used to create 2D and 3D games. It offers cross-platform tools and a large support community.

**Consulta en inglés con contexto:**

"I am a software development student who wants to learn Unity to create an 2D game. What specific benefits does it offer me?"

**Respuesta**: Unity is an excellent choice for 2D games. It allows you to create simple animations and interactive mechanics using C#. It also includes templates to start quickly and a wide library of reusable resources, ideal for projects.

**Consulta traducida al inglés con rol 1:**

"Act as an experienced indie developer. As an indie developer, what advice would you give me for using Unity in my first video game?"

**Respuesta**: If you're starting with Unity, focus first on a small and manageable project. Use the official tutorials and take advantage of the free assets in the Unity Asset Store to speed up your development. Additionally, learn to use the prefab system, which will save you time when creating reusable elements.

**Consulta traducida al inglés con rol 2:**

"Act as a teacher explaining Unity to beginners."

**Respuesta**: Unity is like a toolbox for creating video games. You can design levels, characters, and mechanics using its interface. It is beginner-friendly because you can drag and drop elements, and its programming language, C#, is one of the most popular and accessible ones.

**Conclusión**:

En las dos consultas podemos observar como nos dan unas respuestas similares a las que habíamos obtenido en español, por lo tanto, no observamos diferencias significativas entre idiomas.

## 5. Reformulación de consultas:

**Consulta 1 Original:**

"Cómo influye la elección del motor gráfico en el desarrollo de un videojuego, tanto en términos de calidad como de tiempo de desarrollo?"

**Reformulación:**

"¿Cómo afecta la elección de un motor gráfico específico (como Unreal Engine o Unity) al proceso de desarrollo de un videojuego, considerando tanto la calidad gráfica alcanzable como el tiempo de desarrollo requerido?"

**Respuesta:**

La elección del motor gráfico tiene un impacto significativo en el desarrollo de un videojuego, ya que determina las herramientas y capacidades disponibles:

1. Calidad gráfica:

* Unreal Engine: Es conocido por su potencia gráfica y su capacidad para producir visuales de alta calidad, ideal para juegos AAA y experiencias realistas. Sin embargo, puede requerir más conocimientos técnicos y hardware avanzado.
* Unity: Ofrece flexibilidad, soporta gráficos de alta calidad en 2D y 3D, pero puede necesitar ajustes manuales y complementos para competir con Unreal en visuales hiperrealistas.

1. Tiempo de desarrollo:

* Unreal Engine: Su sistema de blueprints permite reducir el tiempo para prototipos rápidos, pero el aprendizaje inicial puede ser más lento.
* Unity: Es más accesible para principiantes, con una comunidad amplia y recursos como tutoriales. Su flexibilidad también lo hace ideal para proyectos con tiempos ajustados.

1. Adaptabilidad y recursos:

* Unity tiene una curva de aprendizaje más amigable y es preferido por desarrolladores indies por su costo y versatilidad en plataformas móviles. Unreal puede ser más complejo pero es ideal para equipos con recursos avanzados.

En resumen, Unreal destaca en calidad gráfica, pero Unity puede acelerar el desarrollo, especialmente para equipos pequeños o indies.

**Consulta 2 Original:**

"Como desarrollador indie, ¿qué consejos me darías para usar Unity en mi primer videojuego?"

**Reformulación:**

"Actúa como un desarrollador indie experimentado y proporciona una guía práctica para principiantes sobre cómo utilizar Unity eficientemente en su primer videojuego. Considera aspectos como diseño básico, optimización y mejores prácticas."

**Respuesta:**

Como desarrollador indie experimentado, aquí tienes una guía para empezar con Unity en tu primer videojuego:

1. Diseño básico:

* Empieza pequeño: Diseña un juego sencillo con mecánicas básicas. Un plataformas o un runner son buenos puntos de partida.
* Prototipado rápido: Usa las herramientas de Unity, como el Editor y el Sistema de Prefabs, para experimentar con mecánicas antes de comprometerte a un diseño final.

1. Optimización:

* Texturas y modelos: Usa modelos de baja resolución y reduce el tamaño de las texturas para optimizar el rendimiento, especialmente si desarrollas para móviles.
* Scripts eficientes: Evita sobrecargar el juego con actualizaciones innecesarias en tus scripts (usa FixedUpdate con cuidado).
* Prueba en dispositivos reales: Si trabajas para móviles o consolas, asegúrate de probar regularmente en hardware real para optimizar la experiencia.

1. Mejores prácticas:

* Aprende C#: Aunque Unity tiene herramientas visuales, dominar C# te permitirá un mayor control sobre el proyecto.
* Aprovecha los recursos: La Asset Store de Unity tiene modelos, efectos y herramientas gratuitas que pueden ahorrarte tiempo.
* Itera rápido: Haz pruebas frecuentes y obtén retroalimentación temprana. Es mejor corregir errores y ajustar el diseño en las primeras etapas.

En tu primer proyecto, enfócate en aprender la herramienta y terminar un juego funcional más que en crear algo perfecto. La experiencia práctica será tu mejor maestra.

**Conclusión:**

Al comparar las consultas originales con las reformuladas, se observa que las primeras tienden a generar respuestas generales, sin entrar en detalles técnicos ni ofrecer comparaciones relevantes. Por ejemplo, al preguntar sobre motores gráficos, la respuesta inicial menciona su influencia en calidad y tiempo, pero no detalla cómo se diferencian Unity y Unreal Engine, algo crucial para un principiante. De forma similar, los consejos para usar Unity en la pregunta original son vagos y se limitan a sugerencias básicas como "empezar con algo sencillo", lo cual es útil, pero insuficiente.

En cambio, las preguntas reformuladas, al añadir ejemplos concretos o asignar un rol específico, logran guiar mejor las respuestas. En el caso de los motores gráficos, la inclusión de ejemplos como Unity y Unreal permite un análisis comparativo detallado, mostrando sus ventajas y limitaciones. Por otro lado, pedir una guía práctica sobre Unity desde la perspectiva de un desarrollador indie experimentado genera consejos más útiles, como enfocarse en optimización o aprovechar la Asset Store, que son directamente aplicables a proyectos reales.