

Especificación Técnica y Lingüística para el Desarrollo de Sistemas de Corrección Automatizada con Control de Cambios en Entornos OpenXML

1. Introducción y Contextualización del Proyecto

En el actual panorama de la producción editorial y la gestión documental corporativa, la eficiencia en los procesos de revisión y corrección de textos se ha convertido en un imperativo crítico. El proyecto "Antigravity" surge como una respuesta tecnológica a la necesidad de automatizar la fase de pulido ortotipográfico y estilístico, tradicionalmente reservada a la intervención manual de correctores humanos. Sin embargo, la automatización de esta tarea no puede limitarse a la simple sustitución de cadenas de caracteres erróneas; debe integrarse de manera orgánica en los flujos de trabajo profesionales preexistentes. El estándar de facto en la industria para la revisión de documentos es el procesador de textos Microsoft Word y, más específicamente, su funcionalidad de "Control de Cambios" (Track Changes).

La premisa fundamental de este informe es establecer las bases teóricas, normativas y técnicas para el desarrollo de una aplicación capaz de ingerir un manuscrito en formato .docx, analizarlo mediante algoritmos avanzados de procesamiento de lenguaje natural (PLN) y devolver el mismo archivo con las propuestas de corrección marcadas como revisiones pendientes de aprobación. Este enfoque respeta la soberanía del autor sobre su texto, transformando la corrección automática de una imposición destructiva a una sugerencia constructiva.

Para lograr este objetivo, es necesario desglosar la complejidad del problema en tres dimensiones interconectadas: la dimensión lingüística, que define *qué* se debe corregir basándose en la normativa académica y tipográfica vigente; la dimensión estilística, que aborda la calidad y fluidez del texto más allá de la corrección gramatical estricta; y la dimensión tecnológica, que determina *cómo* se implementan dichas correcciones en la estructura XML subyacente de los documentos modernos sin corromper el formato ni la integridad del archivo. A lo largo de este documento, se detallarán exhaustivamente los requerimientos para cada una de estas áreas, proporcionando a Antigravity una hoja de ruta clara para el desarrollo de un motor de corrección de vanguardia.

1.1. El Desafío de la Automatización Editorial

La corrección editorial se divide tradicionalmente en dos grandes áreas: la corrección

ortotipográfica y la corrección de estilo. La primera es objetiva y prescriptiva, regida por normas claras establecidas por instituciones como la Real Academia Española (RAE) y la Asociación de Academias de la Lengua Española (ASALE).¹ La segunda es subjetiva y descriptiva, orientada a mejorar la comunicabilidad del texto según el registro y la intención del autor.²

Un sistema automatizado debe ser capaz de distinguir entre estos dos niveles de intervención. Mientras que un error ortográfico (como la falta de una tilde en una palabra esdrújula) debe ser marcado con un alto nivel de confianza, una sugerencia de estilo (como la recomendación de cambiar una voz pasiva por una activa) debe presentarse con matices, alertando al usuario sobre una posible mejora sin implicar necesariamente una incorrección gramatical.³ La aplicación Antigravity debe, por tanto, operar con una arquitectura híbrida que combine reglas deterministas para la ortotipografía y modelos probabilísticos o heurísticos para el estilo.

Además, la aplicación debe navegar por las tensiones existentes entre las distintas autoridades normativas. Aunque la RAE es la referencia principal para la ortografía léxica, en el ámbito del diseño editorial y la tipografía avanzada, las directrices de autores como José Martínez de Sousa o las recomendaciones de la Fundéu a menudo ofrecen soluciones más refinadas para la composición visual del texto.⁵ El sistema propuesto deberá, idealmente, permitir cierta configurabilidad para adaptarse a "Hojas de Estilo" específicas, eligiendo entre un criterio estrictamente académico o uno más tipográfico.

2. Fundamentos de la Normativa Editorial Hispánica: Ortotipografía

La ortotipografía es la disciplina que aplica las reglas de la ortografía a la tipografía. Es el conjunto de normas que rigen cómo se representan gráficamente los elementos del lenguaje en un texto impreso o digital. Para Antigravity, la implementación rigurosa de estas reglas es lo que diferenciará a la aplicación de un simple corrector ortográfico convencional. A continuación, se detallan las especificaciones lingüísticas que el algoritmo debe verificar y corregir.

2.1. El Sistema de Comillas y su Jerarquía

Uno de los marcadores más distintivos de un texto cuidado en español es el uso correcto de las comillas. A diferencia del inglés, que utiliza principalmente las comillas altas o inglesas (" "), la tradición tipográfica hispánica y la RAE prescriben el uso preferente de las comillas angulares, latinas o españolas (« »).⁷

2.1.1. Jerarquización de Citas Anidadas

El algoritmo debe ser capaz de analizar la estructura de citas dentro del texto y aplicar la jerarquía normativa estricta. Cuando un texto citado contiene a su vez otra cita, se debe seguir el siguiente orden de inclusión:

1. **Nivel Exterior (Primario):** Comillas angulares o latinas (« »).
2. **Nivel Intermedio (Secundario):** Comillas inglesas o altas (" ").
3. **Nivel Interior (Terciario):** Comillas simples (' ').

Lógica de Corrección:

El sistema debe escanear el documento en busca de pares de comillas. Si detecta el uso de comillas inglesas (" ") como nivel primario en un párrafo donde no hay comillas latinas envolventes, debe proponer su sustitución por comillas latinas (« »). Asimismo, si detecta anidamiento, debe verificar que se respete la jerarquía.

- *Caso de Error:* El portavoz declaró: "La situación es 'crítica' según el informe".
- *Corrección Propuesta:* El portavoz declaró: «La situación es "crítica" según el informe».⁷

2.1.2. Puntuación y Comillas

Existe una discrepancia notable entre la norma ortográfica de la RAE y ciertas prácticas tipográficas o influencias de otras lenguas. La RAE establece de manera inequívoca que el punto, la coma, el punto y coma y los dos puntos se escriben siempre *después* de las comillas de cierre, independientemente de si el texto entrecomillado es una oración completa o un fragmento.⁷

- *Regla Algorítmica:* Buscar la secuencia + [Comilla de Cierre] (ej. ." o ,") y proponer la inversión a [Comilla de Cierre] + (ej. ". o ",).
- *Excepciones:* El algoritmo debe ser cauteloso con las llamadas a notas al pie o referencias bibliográficas que puedan aparecer inmediatamente después de la comilla, asegurando que el punto final se coloque tras la llamada de nota si así lo requiere el estilo de citación (aunque la RAE indica que la nota va antes del punto).

2.2. La Raya (—) frente al Guion (-)

Un error omnipresente en los textos no profesionales es el uso del guion corto (hyphen) para funciones que corresponden a la raya (em-dash). La aplicación Antigraffiti debe distinguir semánticamente el uso de estos signos para aplicar la corrección pertinente.

2.2.1. Diálogos y Acotaciones

En los textos narrativos, los diálogos se introducen con una raya. La norma establece que no debe haber espacio entre la raya y el comienzo de la palabra que sigue.

- *Detección:* Párrafos que comienzan con - o — (raya con espacio posterior).
- *Corrección:* Sustituir por — (U+2014) seguido inmediatamente del texto, sin espacio intermedio.¹⁰

2.2.2. Incisos y Oraciones Parentéticas

Cuando se utiliza la raya para introducir una aclaración o inciso dentro de una oración, las reglas de espaciado son específicas: la raya de apertura debe tener un espacio antes y ninguno después; la raya de cierre debe no tener espacio antes y sí tenerlo después (a menos que le siga un signo de puntuación).

- *Incorrecto*: La decisión - arriesgada - fue tomada ayer.
- *Correcto*: La decisión —arriesgada— fue tomada ayer.
- *Incorrecto*: La decisión —arriesgada- fue tomada ayer. (Inconsistencia de signos).

El algoritmo debe identificar pares de guiones o rayas que delimitan una frase y normalizarlos a rayas (U+2014) con el espaciado correcto. Es crucial que el sistema detecte si el inciso termina la oración, en cuyo caso la raya de cierre se omite si coincide con el punto final, o se coloca antes del signo de puntuación si es una coma o punto y coma.¹⁰

2.3. Espaciado Tipográfico y Unidades

La microtipografía se ocupa de los espacios finos y la correcta separación de elementos. Antigraffiti debe implementar reglas para el tratamiento de cifras y símbolos, un área donde la influencia del inglés a menudo introduce errores en textos en español.

2.3.1. Separación de Millares

La norma internacional (ISO) y la académica (RAE) desaconsejan el uso del punto o la coma para separar grupos de tres dígitos en números de más de cuatro cifras, debido a la posible confusión entre decimales y millares en distintos países. La recomendación es usar un espacio fino o un espacio de no separación.¹²

- *Detección*: Patrones numéricos como 10.000 o 10,000.
- *Corrección*: 10 000 (usando U+00A0 No-Break Space o U+202F Narrow No-Break Space).
- *Excepción*: Los años (2024), códigos postales, y leyes no llevan espacio. El algoritmo debe tener contexto para no corregir "año 2025" a "año 2 025".¹⁴

2.3.2. El Símbolo de Porcentaje

A diferencia del inglés, en español se debe dejar un espacio entre la cifra y el símbolo de porcentaje (%). Este espacio debe ser idealmente fino y de no separación para evitar que el símbolo quede huérfano en la siguiente línea.¹⁵

- *Incorrecto*: 100%
- *Correcto*: 100 %
- *Regla Algorítmica*: Buscar (\d)% y reemplazar por \1 % (insertando U+00A0).

Esta misma regla aplica a la mayoría de las unidades de medida (kg, m, km, €), con la excepción notable de los símbolos volados como los grados de ángulo o temperatura cuando

no se especifica la escala (ej. 180°), aunque si se especifica la escala se separa (20 °C).¹⁷

2.4. Uso de Versalitas y Cursivas

La aplicación debe ser capaz de detectar el uso incorrecto de mayúsculas completas en contextos donde la tipografía recomienda versalitas (mayúsculas de tamaño reducido). Esto es común en siglas, números romanos de siglos (en algunas tradiciones tipográficas como la de Martínez de Sousa, aunque la RAE prefiere mayúsculas) y en bibliografías para los apellidos de autores.

- **Siglas:** El uso de versalitas para siglas (ONU, OTAN) es una recomendación estética para evitar "manchas" negras en el texto. Antigravity podría ofrecer esto como una opción de estilo avanzada.
- **Cursivas:** El sistema debe verificar que los extranjerismos crudos (no adaptados) estén marcados en cursiva. Esto requiere contrastar el vocabulario con un diccionario de extranjerismos de la RAE. Si se detecta la palabra "software" o "marketing" en redonda, se debe proponer la cursiva o su equivalente adaptado si existe.¹

3. Especificación Detallada de Reglas de Estilo

Más allá de la corrección normativa, la corrección de estilo busca limpiar el texto de vicios expresivos, redundancias y construcciones sintácticas pobres. Para Antigravity, esto implica un reto mayor, ya que requiere un análisis sintáctico profundo y no solo reconocimiento de patrones superficiales.

3.1. Fenómenos Morfosintácticos

3.1.1. Dequeísmo y Queísmo

El dequeísmo (uso indebido de "de" antes de "que") y el queísmo (supresión indebida de "de") son errores frecuentes que afectan la calidad del texto.

- **Análisis Requerido:** El sistema debe identificar el verbo principal de la oración subordinada y consultar una base de datos de regímenes preposicionales (valencias verbales).
 - Verbo "*pensar*": Transitivo directo. Pienso de que... -> Error (Dequeísmo).
 - Verbo "*acordarse*": Pronominal con régimen preposicional. Me acordé que... -> Error (Queísmo).
- **Implementación:** Se requiere un analizador de dependencias (Dependency Parser) que vincule la cláusula subordinada con el verbo rector para determinar si la preposición es exigida por el verbo o no.¹⁸

3.1.2. Abuso de la Voz Pasiva

El español favorece la voz activa o la pasiva refleja (con "se") sobre la pasiva perifrástica

("ser" + participio), que es más propia del inglés o del lenguaje administrativo anquilosado.

- *Patrón a Detectar*: + [Participio] + [Por (Agente)].
- *Ejemplo*: "La decisión fue tomada por el comité".
- *Propuesta*: "El comité tomó la decisión" (Activa) o "Se tomó la decisión" (Pasiva refleja).
- *Configuración*: Esta regla debe ser configurable, ya que en textos jurídicos o técnicos la pasiva puede ser necesaria. Sin embargo, en textos periodísticos o literarios, su reducción mejora la legibilidad.²⁰

3.1.3. Cosismo y Pobreza Léxica

El "cosismo" es el uso excesivo de la palabra "cosa" para referirse a entidades que tienen nombres más precisos.

- *Algoritmo*: Detectar "cosa/s" cuando funciona como núcleo del sujeto o del objeto directo.
- *Sugerencia Contextual*: Si el verbo es "ocurrir", sugerir "evento" o "suceso". Si el verbo es "decir", sugerir "afirmación" o "comentario". Esto requiere un modelo de lenguaje (LLM) o un tesoro contextual para ofrecer alternativas válidas.²

3.1.4. Redundancias y Pleonasmos

El estilo se depura eliminando palabras innecesarias que no añaden información.

- *Casos Típicos*: "Subir arriba", "bajar abajo", "entrar adentro", "cita previa", "accidente fortuito".
- *Acción*: Marcar la segunda palabra para eliminación (Track Changes: Delete).²³

3.2. Uniformidad y Coherencia (Consistency)

Un aspecto crucial del estilo editorial es la coherencia interna. Antigravity debe escanear el documento completo para detectar inconsistencias en la escritura de términos que admiten variantes.

- *Ejemplo*: Si en la página 1 aparece "Internet" (mayúscula) y en la página 10 "internet" (minúscula), o "psicología" vs. "sicología".
- *Estrategia*: El sistema debe construir un índice de términos variables y determinar cuál es la forma mayoritaria, proponiendo unificar el resto a esa forma dominante, o alertar al usuario para que decida el criterio a seguir.¹⁸

4. Arquitectura de Ingeniería de Software: Manipulación de OpenXML

La implementación técnica de los requerimientos lingüísticos descritos anteriormente presenta un desafío significativo: la necesidad de reflejar las correcciones mediante el control

de cambios de Microsoft Word. La mayoría de las librerías de manipulación de documentos, como la popular python-docx, están diseñadas para leer y escribir el estado *actual* del documento, pero carecen de soporte nativo para generar el marcado XML complejo que Word interpreta como una "revisión" (Track Changes).²⁴

Para que Antigravity cumpla con su promesa de valor, debe operar a un nivel más profundo, manipulando directamente el esquema XML del formato .docx (Office Open XML estándar ECMA-376).

4.1. Anatomía del Archivo.docx

Un archivo .docx es un paquete comprimido (ZIP) que contiene una estructura de directorios y archivos XML. El contenido textual y su marcado se encuentran principalmente en el archivo word/document.xml.

La estructura jerárquica fundamental que Antigravity debe manipular es:

- **<w:document>**: Elemento raíz.
- **<w:body>**: Contenedor del contenido.
- **<w:p> (Paragraph)**: Unidad básica de bloque de texto.
- **<w:r> (Run)**: Fragmento de texto continuo con un mismo formato (negrita, cursiva, etc.).
- **<w:t> (Text)**: El contenido literal del texto.

4.2. El Esquema XML de Control de Cambios (Revision Marking)

Para que Word muestre una corrección como una propuesta (texto tachado rojo para eliminaciones, texto subrayado o coloreado para inserciones), no basta con cambiar el contenido de <w:t>. Se deben envolver los elementos en etiquetas específicas de revisión: <w:ins> (inserción) y <w:del> (eliminación).²⁵

4.2.1. Estructura de Inserción (<w:ins>)

Representa texto que ha sido añadido.

XML

```
<w:ins w:id="0" w:author="Antigravity" w:date="2023-12-21T10:00:00Z">
  <w:r>
    <w:t>texto insertado</w:t>
  </w:r>
</w:ins>
```

- **w:id**: Identificador único de la revisión en el documento. Antigravity debe gestionar un contador para asegurar que no haya colisiones de IDs.

- **w:author:** Debe configurarse como "Antigravity" o el nombre de la app, para que el usuario identifique la fuente de la corrección.
- **w:date:** Marca temporal de la corrección.

4.2.2. Estructura de Eliminación (<w:del>)

Representa texto que ha sido propuesto para borrado.

XML

```
<w:del w:id="1" w:author="Antigravity" w:date="2023-12-21T10:00:00Z">
  <w:r>
    <w:delText>texto borrado</w:delText>
  </w:r>
</w:del>
```

Nótese el uso de <w:delText> en lugar de <w:t> dentro de la estructura de eliminación.

4.2.3. Cambios de Formato (<w:rPrChange>)

Si la corrección implica cambiar formato (ej. poner en cursiva una palabra extranjera), la estructura es diferente. No se borra y escribe texto, sino que se anota el cambio en las propiedades del "Run" (<w:rPr>).

XML

```
<w:r>
  <w:rPr>
    <w:i/> <w:rPrChange w:id="2" w:author="Antigravity" w:date="...">
      <w:rPr/> </w:rPrChange>
    </w:rPr>
  <w:t>software</w:t>
</w:r>
```

Este nivel de detalle es necesario para que el usuario vea el cambio de formato en el panel de revisiones.²⁶

4.3. Estrategias de Implementación Técnica

Dado que python-docx no expone una API de alto nivel para crear estos nodos (paragraph.add_revision() no existe), Antigravity debe implementar una capa de abstracción que combine python-docx para la lectura y navegación básica con lxml para la inyección quirúrgica de nodos XML.²⁸

4.3.1. Habilitación Global del Control de Cambios

Antes de procesar cualquier párrafo, la aplicación debe asegurarse de que el documento tenga activado el seguimiento de revisiones. Esto se hace modificando el archivo word/settings.xml dentro del paquete ZIP.

Se debe inyectar el elemento <w:trackRevisions/> en la configuración. Si este elemento no está presente, aunque inyectemos <w:ins> y <w:del>, Word podría no mostrarlos correctamente o comportarse de forma errática al abrir el archivo.

- *Implementación:* Usar zipfile para abrir el.docx, leer word/settings.xml, parsear con lxml, insertar el tag en la posición correcta (respetando el esquema XSD), y volver a empaquetar.³⁰

4.3.2. Algoritmo de Diferenciación y Reconstrucción (Diffing)

El núcleo lógico de la aplicación será un algoritmo que compare el texto original con el texto corregido y genere las operaciones XML correspondientes.

1. **Extracción:** Obtener el texto plano de un párrafo, pero manteniendo un mapeo de qué caracteres pertenecen a qué <w:r> (Run). Esto es crucial porque un párrafo puede tener múltiples estilos (una palabra en negrita, otra en rojo).
2. **Corrección:** Enviar el texto plano al motor lingüístico (ver sección 5) y recibir la cadena corregida.
3. **Cálculo de Diferencias:** Utilizar una librería como diffliib en Python o la herramienta python-redlines³² para obtener una secuencia de operaciones (igual, insertar, borrar).
4. **Proyección XML (El paso crítico):**
 - Si la operación es "Igual": Se conserva el <w:r> original.
 - Si la operación es "Borrar": Se envuelve el <w:r> (o la parte correspondiente de él) en un <w:del>. Si el borrado abarca varios Runs con distintos formatos, se deben generar múltiples nodos <w:del> para mantener la validez del XML.
 - Si la operación es "Insertar": Se crea un nuevo <w:ins> conteniendo un nuevo <w:r> con el texto nuevo. El formato de este nuevo Run debe heredarse del contexto adyacente para mantener la coherencia visual.³⁴

4.3.3. Uso de Librerías Existentes: python-redlines

Para acelerar el desarrollo, Antigravity puede evaluar el uso de python-redlines. Esta librería actúa como un wrapper que simplifica la creación de XML con marcas de revisión.

- *Ventaja:* Abstrae la complejidad de w:ins y w:del. Produce un XML válido compatible con Word.
- *Limitación:* Es fundamental verificar si soporta la preservación de formatos complejos

(tablas, imágenes anidadas, notas al pie). Si la librería reconstruye el documento desde cero basándose solo en texto, podría perderse información no textual. En tal caso, sería preferible una implementación propia usando lxml sobre la estructura original.³³

5. Integración de Motores de Procesamiento de Lenguaje Natural (PLN)

El cerebro de Antigravity reside en su capacidad para detectar errores. Para ello, se recomienda una arquitectura compuesta que integre múltiples motores de análisis.

5.1. LanguageTool como Motor Base

LanguageTool es la solución de código abierto más robusta para la corrección gramatical y de estilo en español. Su arquitectura basada en reglas permite una personalización profunda.³⁶

5.1.1. Configuración de Reglas

Para cumplir con los requerimientos de estilo y ortotipografía, no basta con la configuración por defecto. Antigravity debe implementar un perfil de reglas personalizado:

- **Reglas a Activar:**
 - PASSIVE_VOICE (ID: PASSIVE_VOICE): Para detectar construcciones pasivas perifrásticas. Por defecto suele estar desactivada o en modo "picky".³⁷
 - DE_QUE (ID: DE_QUE): Reglas específicas para detectar dequeísmo y queísmo.¹⁹
 - WHITESPACE_RULE: Para detectar dobles espacios o espacios faltantes antes de unidades.³⁹
 - SMART_QUOTES: Para forzar el uso de comillas tipográficas (latinas).
- **Reglas a Desactivar:** Reglas que generen ruido o falsos positivos en contextos literarios o técnicos específicos.
- **Archivo grammar.xml Personalizado:** Se recomienda crear un archivo grammar_custom.xml que contenga reglas específicas para los requerimientos de Antigravity (ej. detección de "cosismo" si no está cubierto nativamente) y cargarlo en la instancia del servidor LanguageTool.⁴⁰

5.1.2. Interfaz Técnica

Se recomienda utilizar la librería language_tool_python para interactuar con una instancia local del servidor LanguageTool (escrito en Java). Esto evita latencias de red y problemas de privacidad de datos, ya que el texto del usuario no sale de la infraestructura de la app.³⁶

5.2. SpaCy para Análisis Profundo

Para detecciones que requieren comprensión de la estructura arbórea de la oración (sintaxis), LanguageTool puede quedarse corto. Aquí entra SpaCy con sus modelos es_core_news_lg o

trf (Transformer).

- **Detección de Cosismo:** Mediante el análisis de dependencias, se puede identificar cuando el lema "cosa" actúa como núcleo de un complemento directo o sujeto, y analizar el verbo rector para sugerir un sinónimo más preciso (ej. "decir una cosa" -> "hacer una afirmación").⁴²
- **Concordancia a Larga Distancia:** Verificar la concordancia sujeto-verbo cuando hay oraciones subordinadas interpuestas, algo que los correctores simples a menudo pasan por alto.

6. Recomendaciones de Diseño de Experiencia de Usuario (UX) en Word

El éxito de la aplicación dependerá de cómo el usuario final perciba las correcciones dentro de Word.

6.1. Visualización de Cambios

Al abrir el documento procesado por Antigravity, el usuario debe ver inmediatamente el panel de revisiones.

- **Recomendación:** El archivo debe guardarse con la configuración de vista "Mostrar marcas" activada por defecto en los metadatos de configuración (settings.xml), para que el usuario no tenga que activar el control de cambios manualmente para ver lo que hizo la app.
- **Autoría:** Todos los cambios deben aparecer bajo un único usuario "Antigravity" (o "Corrector Automático"). Esto permite al usuario filtrar y aceptar/rechazar en bloque si lo desea.

6.2. Comentarios vs. Cambios

Para las correcciones de ortotipografía (objetivas), el uso de Track Changes (sustitución directa) es ideal.

Para las correcciones de estilo (subjetivas), donde la sustitución directa puede ser intrusiva, se recomienda usar la inserción de Comentarios (<w:comment>).

- *Implementación:* En lugar de cambiar "La cosa es que..." por "La situación es que...", la app podría insertar un comentario en la selección: "Sugerencia de estilo: Considere sustituir 'cosa' por un término más preciso como 'situación' o 'problema'".
- Esto requiere manipular el archivo word/comments.xml y vincularlo al texto mediante <w:commentRangeStart> y <w:commentRangeEnd> en document.xml.⁴⁴ Esta aproximación híbrida (Cambios para errores, Comentarios para estilo) ofrece una experiencia profesional superior.

7. Conclusiones y Hoja de Ruta de Implementación

El desarrollo de la aplicación de corrección para Antigravity es un proyecto viable que requiere una integración cuidadosa de tecnologías de ingeniería de software (manipulación XML) y lingüística computacional. No existe una "bala de plata" o librería única que resuelva todo el problema "out-of-the-box"; la solución reside en la orquestación de lxml para la estructura del documento y LanguageTool/SpaCy para la inteligencia lingüística.

Resumen de Requerimientos Clave:

Área	Requerimiento Principal	Solución Técnica Propuesta
Formato	Procesar.docx manteniendo estilos, imágenes y tablas.	python-docx para lectura + lxml para inyección de revisiones.
Output	Marcar cambios usando Track Changes nativo de Word.	Inyección de nodos <w:ins>, <w:del> y <w:pPrChange>.
Normativa	Aplicar reglas RAE y tipográficas (comillas, rayas).	Motor de reglas Regex + LanguageTool personalizado.
Estilo	Detectar pasiva, dequeísmo, cosismo.	Análisis morfosintáctico con SpaCy + Reglas gramaticales.
UX	Diferenciar corrección obligatoria de sugerencia.	Uso híbrido de Revisiones (Track Changes) y Comentarios.

Hoja de Ruta (Roadmap) Sugerida:

1. **Fase 1: Prototipo de Infraestructura.** Crear un script en Python capaz de leer un.docx, activar "Track Changes" en su XML, realizar una sustitución de texto simple (ej. "a" por "b") inyectando los nodos w:ins/w:del correctamente, y generar un archivo que Word abra sin errores.

2. **Fase 2: Motor Ortotipográfico.** Implementar las reglas de Regex para comillas, rayas y espacios. Integrarlas en el flujo de la Fase 1.
3. **Fase 3: Integración NLP.** Conectar la API de LanguageTool para detectar errores gramaticales y mapear sus offsets a los nodos XML del documento.
4. **Fase 4: Refinamiento de Estilo.** Desarrollar módulos específicos para "Cosismo" y "Voz Pasiva" usando SpaCy, decidiendo si se presentan como cambios o comentarios.

Siguiendo estas recomendaciones, Antigravity podrá desplegar una herramienta de corrección que no solo compite con los correctores estándar, sino que ofrece un valor añadido profesional alineado con las exigencias del sector editorial de alto nivel.

Obras citadas

1. Corrección ortográfica y de estilo: ¿en qué se diferencian y por qué ..., fecha de acceso: diciembre 21, 2025, <https://cursiva.com/blog/news/correccion-ortografica-y-de-estilo-en-que-se-diferencian-y-por-que-son-esenciales>
2. Corrección ortotipográfica o de estilo - Blog - Editorial Círculo Rojo, fecha de acceso: diciembre 21, 2025, <https://blog.editorialcirculo rojo.com/blog/correccion-ortotipografica-o-de-estilo>
3. Corrección ortotipográfica y de estilo - Letras Propias, fecha de acceso: diciembre 21, 2025, <https://letraspropias.com/correccion-ortotipografica-y-de-estilo/>
4. Diferencia entre la corrección de estilo y la corrección ortotipográfica, fecha de acceso: diciembre 21, 2025, <https://revengaediciones.com/diferencia-entre-la-correccion-de-estilo-y-la-correccion-ortotipografica/>
5. de Sousa | La Piedra Translatofal, fecha de acceso: diciembre 21, 2025, <https://lapiedratranslatofal.wordpress.com/tag/de-sousa/>
6. Signos ortográficos, ortotipografía y normas actuales, fecha de acceso: diciembre 21, 2025, https://www.uv.es/normas/2012/ANEJOS/Signos_2012.pdf
7. comillas | Diccionario panhispánico de dudas | RAE - ASALE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/dpd/comillas>
8. Usos y Tipos de Comillas en Español | PDF - Scribd, fecha de acceso: diciembre 21, 2025, <https://es.scribd.com/document/807821558/3-Las-Comillas-en-El-DPD-de-La-RAE-PE>
9. Las comillas | El buen uso del español | RAE - ASALE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/buen-uso-esp%C3%B1ol/las-comillas>
10. 3.2.7 la raya | Ortografía básica de la lengua española | RAE - ASALE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/ortograf%C3%ADa-b%C3%A1sica/uso-de-los-signos-ortogr%C3%A1ficos/signos-de-puntuaci%C3%B3n/la-raya>
11. La raya - Correcciones Santaella, fecha de acceso: diciembre 21, 2025, <https://correccionessantaella.es/2014/05/12/la-raya/>

12. miles y millones, claves de escritura - Fundéu, fecha de acceso: diciembre 21, 2025, <https://www.fundeu.es/recomendacion/miles-y-millones-claves-de-escritura/>
13. números | Diccionario panhispánico de dudas | RAE - ASALE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/dpd/n%C3%BAmeros>
14. Uso de palabras o cifras en la escritura de la fecha | RAE - ASALE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/ortograf%C3%ADa/uso-de-palabras-o-cifras-en-la-escritura-de-la-fecha>
15. La expresión de los porcentajes | El buen uso del español | RAE, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/buen-uso-espa%C3%B1ol/la-expresi%C3%B3n-de-los-porcentajes>
16. ¿El símbolo del porcentaje se escribe pegado a la cifra?, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/duda-linguistica/el-simbolo-del-porcentaje-se-escribe-pegado-a-la-cifra>
17. ¿Los símbolos se escriben pegados a la cifra? | Duda lingüística, fecha de acceso: diciembre 21, 2025, <https://www.rae.es/duda-linguistica/los-simbolos-se-escriben-pegados-a-la-cifra>
18. Corrección de estilo y ortotipográfica - Editorial Letra Minúscula, fecha de acceso: diciembre 21, 2025, <https://www.letraminuscula.com/correccion-de-estilo-y-ortotipografica/>
19. Rule "de que" - LanguageTool Community (), fecha de acceso: diciembre 21, 2025, https://community.languagetool.org/rule/show/DE_QUE?lang=ca-ES&subld=12
20. Rule "Passive voice" - LanguageTool Community (), fecha de acceso: diciembre 21, 2025, https://community.languagetool.org/rule/show/PASSIVE_VOICE?lang=en&subld=6
21. Rule "Passive voice" - LanguageTool Community (), fecha de acceso: diciembre 21, 2025, https://community.languagetool.org/rule/show/PASSIVE_VOICE?lang=en&subld=1
22. Lenguaje Claro en la Administración | PDF | Lingüística - Scribd, fecha de acceso: diciembre 21, 2025, <https://es.scribd.com/document/769744186/Lenguaje-Claro-y-tecnologia-en-la-administracion>
23. LanguageTool as a CAT tool for Easy-to-Read in Spanish, fecha de acceso: diciembre 21, 2025, <https://aclanthology.org/2024.readi-1.8.pdf>
24. How to Accept ("Preserve") DOCX Commented Changes In ... - GitHub, fecha de acceso: diciembre 21, 2025, <https://github.com/python-openxml/python-docx/issues/566>
25. docx: partitioner finds text nested in revision-marks #1821 - GitHub, fecha de acceso: diciembre 21, 2025, <https://github.com/Unstructured-IO/unstructured/issues/1821>
26. ParagraphPropertiesChange Class (DocumentFormat.OpenXml ..., fecha de acceso: diciembre 21, 2025,

- <https://learn.microsoft.com/en-us/dotnet/api/documentformat.openxml.wordprocessing.paragraphpropertieschange?view=openxml-3.0.1>
27. ParagraphMarkRunPropertiesCh..., fecha de acceso: diciembre 21, 2025, <https://learn.microsoft.com/en-us/dotnet/api/documentformat.openxml.wordprocessing.paragraphmarkrunpropertieschange?view=openxml-3.0.1>
 28. Python-docx: Is it possible to add a new run to paragraph in a ..., fecha de acceso: diciembre 21, 2025, <https://stackoverflow.com/questions/52740630/python-docx-is-it-possible-to-add-a-new-run-to-paragraph-in-a-specific-place-n>
 29. How do I Add Custom XML to An Element Using Python-Docx?, fecha de acceso: diciembre 21, 2025, <https://stackoverflow.com/questions/70992530/how-do-i-add-custom-xml-to-an-element-using-python-docx>
 30. Track Changes in Word DOC using Python - Aspose Knowledge Base, fecha de acceso: diciembre 21, 2025, <https://kb.aspose.com/words/python/track-changes-in-word-doc-using-python/>
 31. How to activate "Track changes"-Mode in a word document ..., fecha de acceso: diciembre 21, 2025, <https://stackoverflow.com/questions/67061294/how-to-activate-track-changes-mode-in-a-word-document-generated-with-python>
 32. Python Tool for Docx Tracked Change Comparisons - Reddit, fecha de acceso: diciembre 21, 2025, https://www.reddit.com/r/Python/comments/198wuja/python_tool_for_docx_tracked_change_comparisons/
 33. redlines - PyPI, fecha de acceso: diciembre 21, 2025, <https://pypi.org/project/redlines/>
 34. Highlight inserted,deleted elements/text - Python Docx, fecha de acceso: diciembre 21, 2025, <https://stackoverflow.com/questions/75298403/highlight-inserted-deleted-elements-text-python-docx>
 35. JSv4/Python-Redlines: Docx tracked change redlines for ... - GitHub, fecha de acceso: diciembre 21, 2025, <https://github.com/JSv4/Python-Redlines>
 36. jxmorris12/language_tool_python: a free python grammar checker, fecha de acceso: diciembre 21, 2025, https://github.com/jxmorris12/language_tool_python
 37. Rule "Passive voice" - LanguageTool Community (), fecha de acceso: diciembre 21, 2025, https://community.languagetool.org/rule/show/PASSIVE_VOICE?lang=en&subId=5
 38. Issue #9 · dpelle/vim-LanguageTool - Passive voice rule - GitHub, fecha de acceso: diciembre 21, 2025, <https://github.com/dpelle/vim-LanguageTool/issues/9>
 39. Browse LanguageTool Rules: 6,149 matches for English, fecha de acceso: diciembre 21, 2025, <https://community.languagetool.org/rule/list?lang=en>
 40. Developing Robust Rules - dev.languagetool.org, fecha de acceso: diciembre 21, 2025, <https://dev.languagetool.org/developing-robust-rules.html>
 41. languagetool-language-modules - es - src - main - resources - GitHub, fecha de acceso: diciembre 21, 2025,

- <https://github.com/languagetool-org/languagetool/blob/master/languagetool-language-modules/es/src/main/resources/org/languagetool/rules/es/grammar.xml>
42. spaCy · Industrial-strength Natural Language Processing in Python, fecha de acceso: diciembre 21, 2025, <https://spacy.io/>
 43. PROCESAMIENTO DE LENGUAJE NATURAL PARA LAS ..., fecha de acceso: diciembre 21, 2025, https://www.researchgate.net/profile/Christian-Maldonado-Sifuentes/publication/355378872_PROCESAMIENTO_DE_LENGUAJE_NATURAL_PARA_LAS_LENGUAS_INDIGENAS/links/616d96ad951b3574c662e998/PROCESAMIENTO-DE-LENGUAJE-NATURAL-PARA-LAS-LENGUAS-INDIGENAS.pdf
 44. feature: insert comment · Issue #93 · python-openxml/python-docx, fecha de acceso: diciembre 21, 2025, <https://github.com/python-openxml/python-docx/issues/93>
 45. Document objects - python-docx - Read the Docs, fecha de acceso: diciembre 21, 2025, <https://python-docx.readthedocs.io/en/latest/api/document.html>