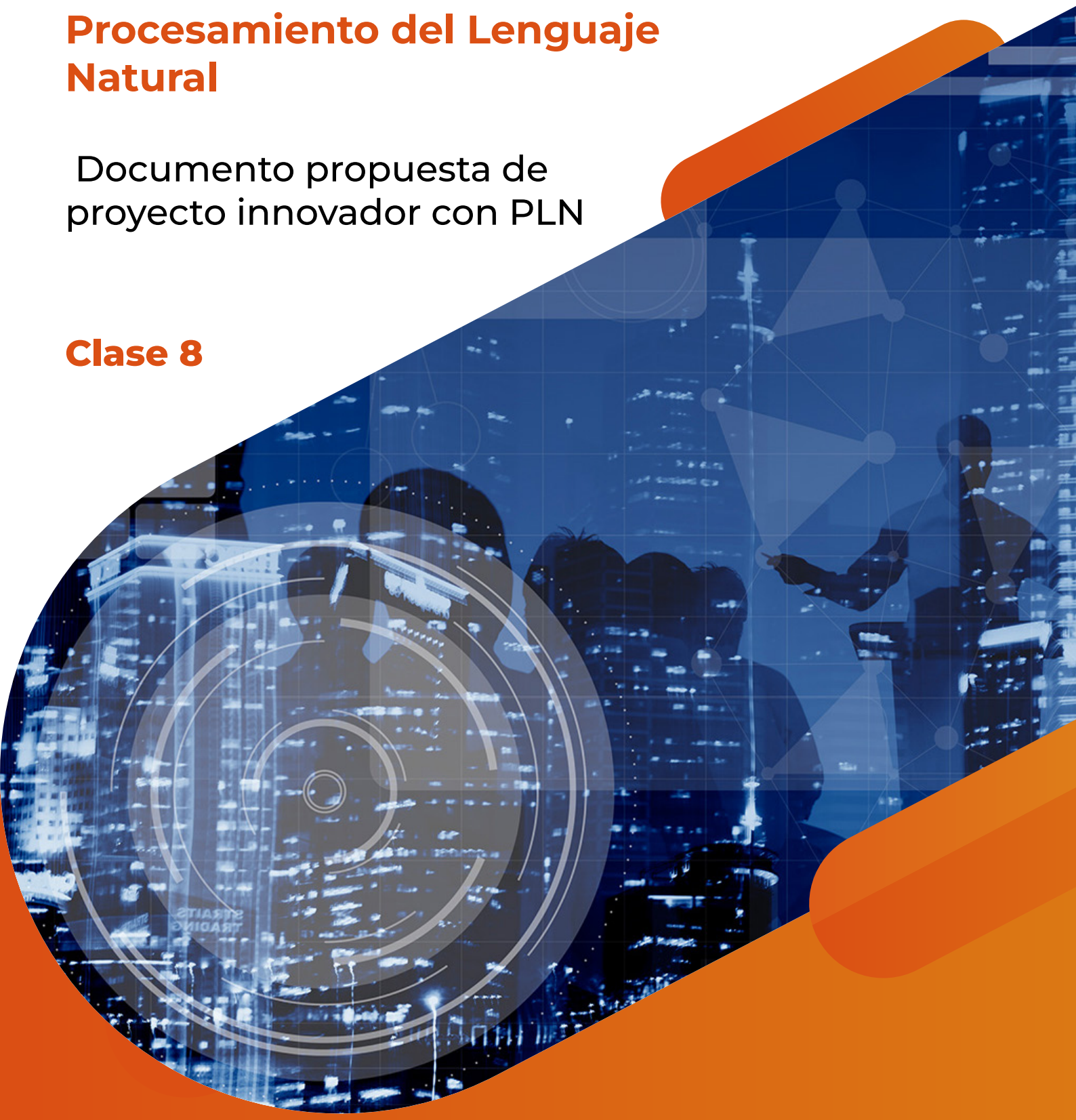
PUCE Virtual
EDUCACIÓN FLEXIBLE

Procesamiento del Lenguaje Natural

Documento propuesta de proyecto innovador con PLN

Clase 8



Maestria en Educacion en Inteligencia Artificial y Entornos Virtuales

La excelencia no se improvisa

Introducción

En esta clase se aborda el Procesamiento de Lenguaje Natural (PLN) como una herramienta transformadora en el ámbito educativo. El enfoque principal está en diseñar un proyecto innovador que integre PLN para mejorar prácticas educativas. Se exploran problemáticas clave, como la comprensión lectora, la retroalimentación automatizada y el análisis de desempeño estudiantil, con el objetivo de crear entornos de aprendizaje más inclusivos, efectivos y adaptativos. Además, se destaca el papel del PLN en la optimización de la gestión del conocimiento, el fomento del pensamiento crítico y el fortalecimiento de la alfabetización digital.

En cuanto a los resultados de aprendizaje, los estudiantes evaluarán sistemas de PLN para su utilización en ambientes educativos, desarrollarán habilidades para formular objetivos claros y alcanzables en proyectos educativos con PLN, definirán metodologías robustas que integren enfoques cualitativos y cuantitativos, y seleccionarán herramientas tecnológicas adecuadas. También adquirirán competencias para diseñar soluciones pedagógicas y tecnológicas alineadas con necesidades educativas específicas, considerando criterios éticos, de escalabilidad y sostenibilidad en la implementación de proyectos educativos basados en PLN.

Clase 8

15. Documento propuesta de proyecto innovador con PLN

En el ámbito educativo, el Procesamiento de Lenguaje Natural (PLN) ofrece oportunidades para transformar los procesos de enseñanza y aprendizaje, al facilitar la personalización de contenidos, el análisis de grandes volúmenes de información y la mejora de la interacción entre docentes y estudiantes. Esta clase se enfoca en la elaboración de un proyecto que proponga el diseño de una solución innovadora basada en PLN, orientada a mejorar prácticas educativas a través de tecnologías emergentes. Al abordar problemáticas como la comprensión lectora, la retroalimentación automatizada y el análisis de desempeño estudiantil, se busca demostrar cómo el PLN puede contribuir a entornos de aprendizaje más inclusivos, efectivos y adaptativos. La elaboración de la propuesta se fundamentará en el potencial del PLN para optimizar la gestión del conocimiento, potenciar la alfabetización digital y fomentar el pensamiento crítico en el alumnado. En este sentido, se explorarán estrategias que permitan integrar el PLN en procesos educativos, promoviendo así una educación más inteligente, flexible y centrada en el estudiante. El desarrollo y gestión de proyectos tiene muchas similitudes entre diversos campos, por lo que se recomienda ver este video general sobre el tema:

https://youtu.be/Og7iI31kwPQ?si=ALeyIhM190RAoaD6.

15.1. Objetivos

Los objetivos son elementos fundamentales en cualquier proyecto, ya que proporcionan dirección, delimitan el alcance y sirven como base para la evaluación de resultados. En proyectos de PLN aplicados al ámbito educativo, la formulación de objetivos requiere un enfoque técnico que contemple tanto las capacidades tecnológicas del PLN como las necesidades educativas específicas. Esta sección proporciona una guía paso a paso para la elaboración de objetivos claros, medibles y alcanzables (Figura 1), con ejemplos prácticos que ilustran su aplicación.

Figura 1. Definición de objetivos

Nota: Creado con DALL-E (2025)

15.1.1. Comprensión del propósito del proyecto

Antes de definir los objetivos, es esencial tener claridad sobre el propósito del proyecto. En el contexto educativo, los proyectos de PLN pueden perseguir fines como:

- Mejorar la comprensión lectora mediante el análisis semántico de textos.
- Desarrollar sistemas de tutoría inteligente que proporcionen retroalimentación personalizada.
- Implementar herramientas de análisis de sentimientos para evaluar el bienestar emocional de los estudiantes.

Ejemplo:

Propósito del proyecto: Desarrollar una plataforma que utilice PLN para mejorar la comprensión lectora en estudiantes de secundaria mediante análisis automático de textos.

15.1.2. Definición del objetivo general

El objetivo general describe el propósito global del proyecto, de forma clara y amplia, sin detallar los medios para alcanzarlo. Debe responder a la pregunta: ¿Qué se pretende lograr? (Figura 2).

Características:

- Claridad y concreción.
- Coherencia con el propósito general.

- Enfoque en resultados.

Ejemplo:

***Objetivo general**:* Diseñar e implementar una plataforma educativa basada en PLN que optimice la comprensión lectora en estudiantes de secundaria a través del análisis automático de textos académicos.

Figura 2. Objetivo general

Nota: Creado con DALL-E (2025)

15.1.3. Formulación de objetivos específicos

Los objetivos específicos detallan las acciones necesarias para alcanzar el objetivo general. Cada objetivo debe ser:

- Específico: Indicar con claridad lo que se hará.
- Medible: Determinar cómo se evaluará su cumplimiento.
- Alcanzable: Ser viable con los recursos disponibles.
- Relevante: Contribuir directamente al objetivo general.
- Temporal: Establecer un marco de tiempo.

Ejemplos de objetivos específicos:

1. Analizar corpus de textos académicos relevantes para identificar patrones lingüísticos asociados a la comprensión lectora.
2. Desarrollar algoritmos de PLN para el análisis semántico y sintáctico de textos.
3. Implementar un sistema de retroalimentación automática que sugiera estrategias de lectura personalizadas.
4. Evaluar la eficacia de la plataforma mediante pruebas piloto en instituciones educativas.

15.1.4. Uso de verbos de acción

La correcta elección de verbos garantiza precisión y claridad. Para proyectos de PLN en educación, se recomiendan verbos como:

- Analizar: Identificar patrones lingüísticos en textos educativos.

- Diseñar: Crear modelos de PLN para el procesamiento de lenguaje natural.
- Implementar: Poner en funcionamiento la solución tecnológica propuesta.
- Evaluar: Medir la efectividad del sistema en contextos educativos.

Ejemplo:

Objetivo específico: Diseñar un modelo de PLN que detecte errores semánticos y gramaticales en redacciones estudiantiles.

15.1.5. Alineación con indicadores de evaluación

Cada objetivo debe vincularse con indicadores que permitan medir su cumplimiento.

Ejemplo:

- *Objetivo específico:* Implementar algoritmos de PLN para evaluar la coherencia de ensayos estudiantiles.
- *Indicador:* Porcentaje de ensayos evaluados automáticamente con un margen de error inferior al 5% en comparación con evaluaciones manuales.

15.2. Metodología

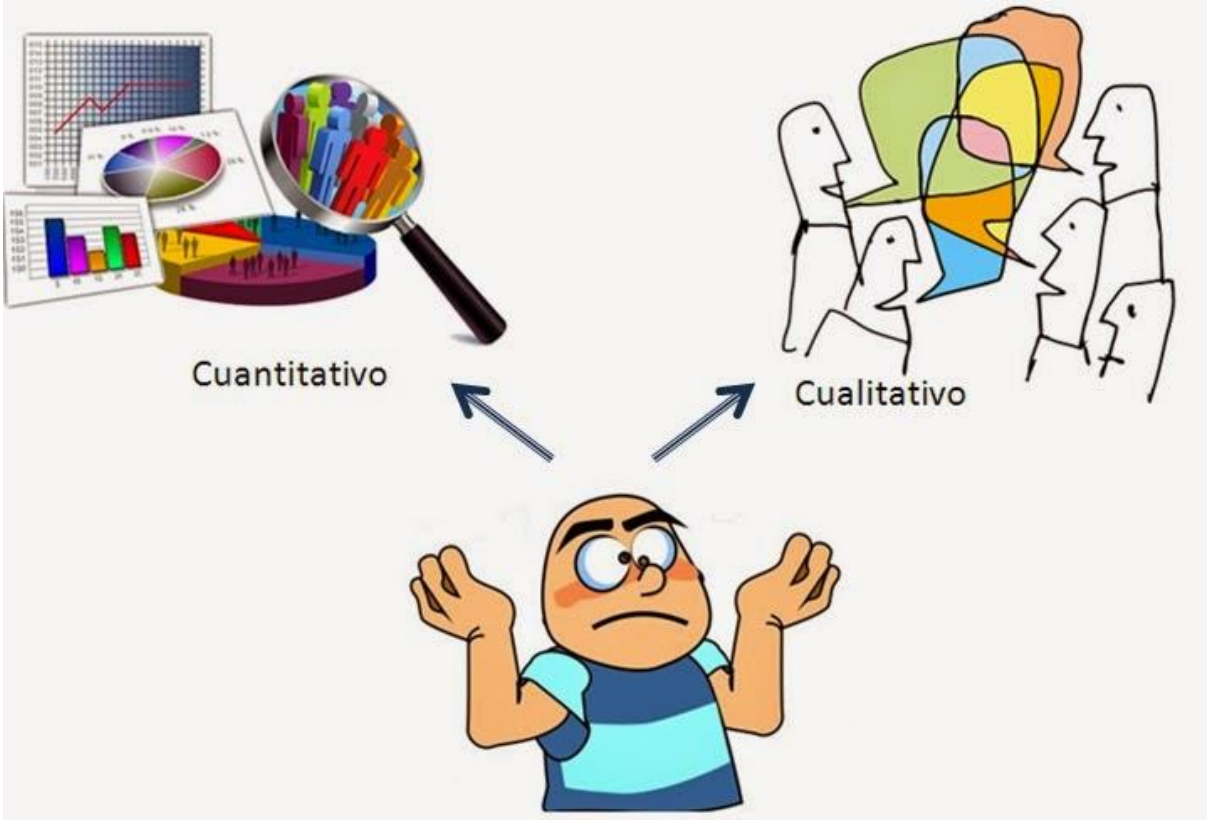
Esta sección es clave, ya que define los enfoques, técnicas y herramientas utilizadas para alcanzar los objetivos propuestos. Una metodología bien estructurada garantiza la viabilidad técnica del proyecto, facilita su replicabilidad y asegura resultados robustos y medibles. Esta sección proporciona una guía detallada para elaborar la metodología, incluyendo información técnica y ejemplos prácticos. Un grupo de metodologías de gran uso actualmente, especialmente en el desarrollo de software, son las metodologías ágiles, que pueden ser una gran alternativa para proyectos de IA: https://youtu.be/Smf2BKwiBmM?si=K_yuK1yH1hkUtDiP.

15.2.1. Selección del enfoque metodológico

El primer paso es definir el enfoque general que guiará el proyecto. Para proyectos de PLN, los enfoques más comunes son (Figura 3):

- **Cualitativo:** Orientado al análisis profundo de textos para identificar patrones lingüísticos y semánticos.
- **Cuantitativo:** Centrado en el uso de algoritmos para procesar grandes volúmenes de datos lingüísticos.
- **Mixto:** Combina ambos enfoques para obtener una comprensión integral del problema educativo.

Figura 3. Metodologías de investigación



Nota: Tomado de Brito (2015)

Ejemplo:

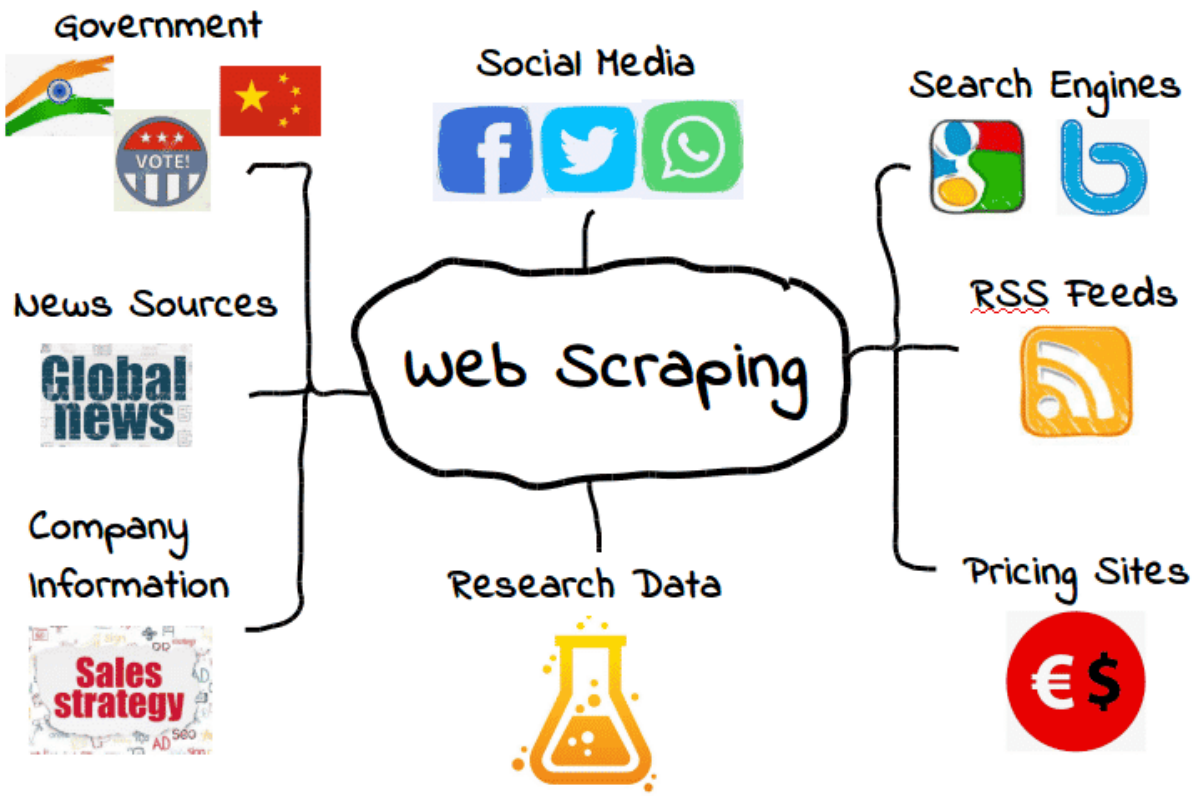
El proyecto adoptará un enfoque mixto para analizar corpus de textos académicos, utilizando aprendizaje automático para el procesamiento y entrevistas a docentes para validar los resultados.

15.2.2. Definición de las etapas del proyecto

Una metodología clara debe descomponer el proyecto en etapas. A continuación, se describen las principales fases:

- Recolección de Datos
    - Descripción: Reunir corpus de textos educativos, respuestas estudiantiles o datos de interacción en plataformas de aprendizaje.
    - Herramientas: Web scraping (Figura 4), APIs educativas, bases de datos lingüísticas.

Figura 4. Web scraping



Nota: Tomado de Kinsta (2022)

Ejemplo:

Se recopilarán textos de libros de secundaria mediante técnicas de web scraping, garantizando el cumplimiento de normativas de derechos de autor.

- Preprocesamiento de Datos
    - Descripción: Limpieza y normalización del lenguaje, incluyendo eliminación de ruido, tokenización, lematización y corrección gramatical.
    - Herramientas: NLTK (NLTK, 2024), SpaCy (Explosion, 2025), pandas (NumFOCUS, 2024).

Ejemplo:

Los textos recopilados serán preprocesados con SpaCy para garantizar consistencia en el análisis semántico.

- **Desarrollo del modelo de PLN**
    - Descripción: Selección y entrenamiento de modelos de PLN (por ejemplo, redes neuronales recurrentes o transformadores como BERT) para tareas específicas, como análisis de sentimientos o clasificación de texto.
    - Herramientas: TensorFlow (TensorFlow, 2025a), PyTorch (PyTorch, 2025), Hugging Face (Hugging Face, 2025).

Ejemplo:

Se entrenará un modelo BERT adaptado al español para evaluar la coherencia en ensayos estudiantiles.

- **Evaluación y validación**
    - Descripción: Validar el modelo utilizando métricas estándar (precisión, recall, F1-score) y pruebas piloto en entornos educativos.
    - Herramientas: Scikit-learn (Scikit-learn, 2025), evaluaciones comparativas.

Ejemplo:

La plataforma se validará con una muestra piloto de 200 estudiantes, comparando los resultados del modelo con evaluaciones realizadas por docentes.

- Implementación y escalabilidad
    - Descripción: Despliegue del sistema en plataformas educativas, considerando aspectos de escalabilidad y mantenimiento.

Ejemplo:

La plataforma se implementará en un entorno de nube con escalabilidad automática para ajustarse a la demanda institucional.

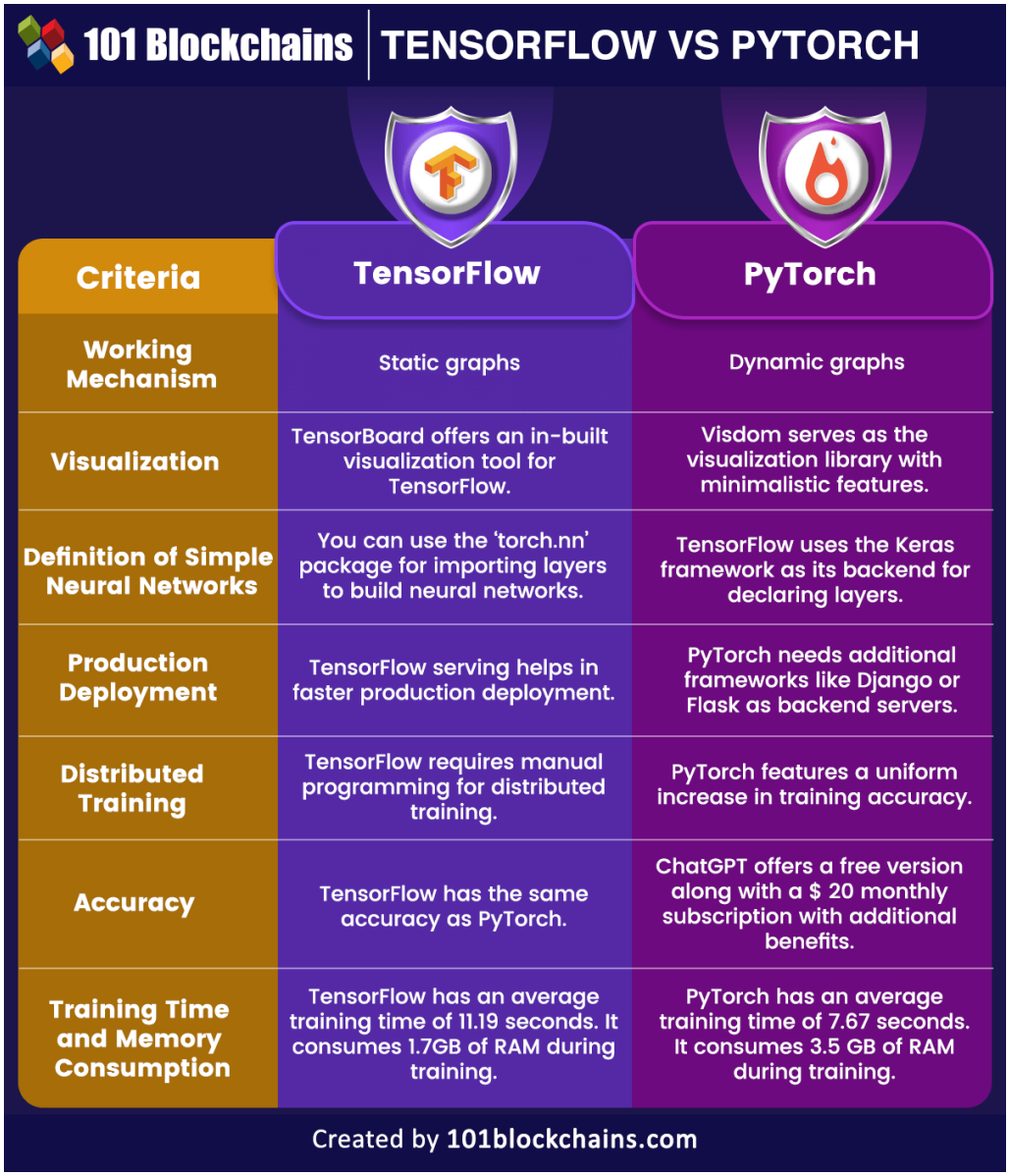
15.2.3. Elección de herramientas y tecnologías

La selección de tecnologías debe alinearse con los objetivos y el alcance del proyecto.

Recomendaciones Técnicas:

- Lenguaje de programación: Python, por su robustez en bibliotecas de PLN.
- Plataformas: Google Cloud (Google, 2025a), AWS (Amazon, 2025a) para implementación escalable.
- Frameworks: TensorFlow, PyTorch (Figura 5) para desarrollo de modelos.

Figura 5. Tensorflow vs. PyTorch



Nota: Tomado de Weston (2023)

Ejemplo:

El proyecto utilizará Python y TensorFlow para el desarrollo del modelo, implementado en Google Cloud para garantizar su disponibilidad y escalabilidad.

15.2.4. Gestión de riesgos y consideraciones éticas

- Riesgos técnicos: Precisión insuficiente del modelo, sesgos en los datos.
- Riesgos éticos: Protección de datos estudiantiles, sesgos algorítmicos.

Estrategias de mitigación (Figura 6):

- Implementación de pruebas continuas de sesgo.
- Cumplimiento de regulaciones de privacidad de datos.

Figura 6. Gestión de riesgos

Nota: Tomado de Amblard-Ladurantie (2024)

Ejemplo:

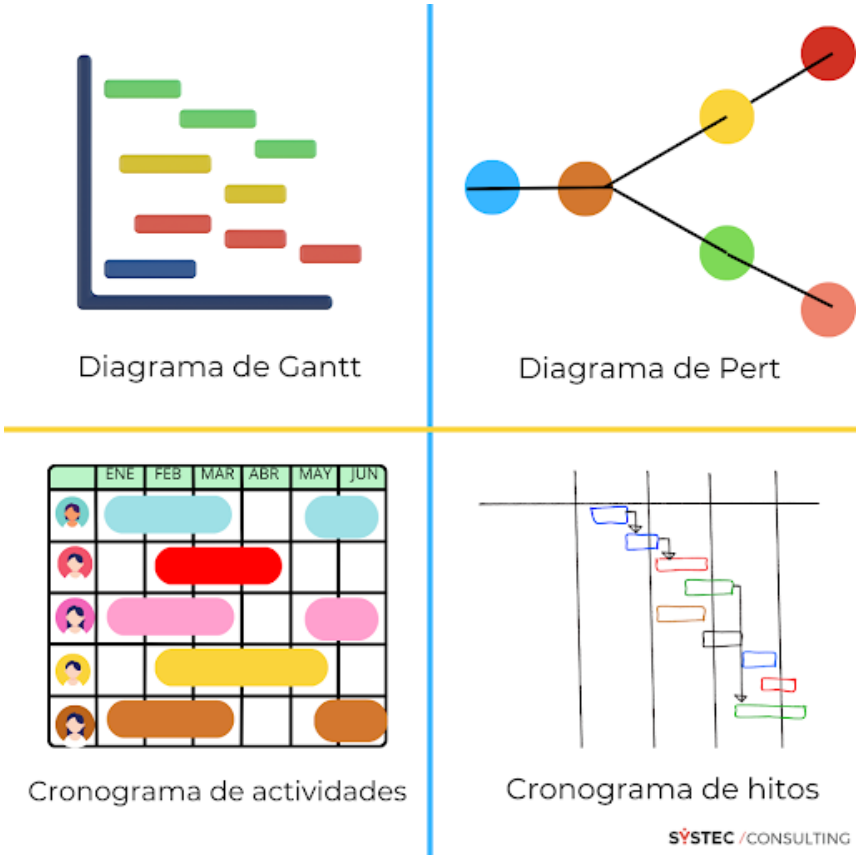
Se aplicarán técnicas de detección de sesgos en modelos para asegurar la equidad en la evaluación de textos estudiantiles.

15.2.5. Cronograma y asignación de recursos

Un cronograma detallado facilita el seguimiento del proyecto (Figura 7). Debe incluir:

- Tiempos estimados para cada fase.
- Asignación de roles y recursos humanos.
- Identificación de hitos clave.

Figura 7. Cronograma de actividades



Nota: Tomado de Rodríguez (2024)

Ejemplo:

La fase de desarrollo del modelo se estima en 8 semanas, con un equipo compuesto por un científico de datos, un lingüista computacional y un especialista en educación.

15.3. Diseño

El diseño de un proyecto de PLN en el ámbito educativo es una fase clave que define la estructura, arquitectura y funcionamiento de la solución tecnológica. Esta etapa asegura que todos los componentes del proyecto estén alineados con los objetivos y la metodología propuesta, garantizando su viabilidad técnica y pertinencia pedagógica. Esta sección describe el proceso para elaborar un diseño sólido y escalable, incluyendo componentes técnicos, arquitectónicos y pedagógicos, junto con ejemplos específicos.

15.3.1. Definición del alcance y requerimientos del proyecto

Antes de diseñar, se deben especificar claramente los requerimientos del sistema, tanto funcionales como no funcionales (Figura 8).

- Requerimientos funcionales: Describen lo que el sistema debe hacer, por ejemplo, análisis de sentimientos en ensayos estudiantiles o generación automática de resúmenes.
- Requerimientos no funcionales: Involucran aspectos como la escalabilidad, seguridad, rendimiento y accesibilidad de la solución.

Figura 8. Requerimientos funcionales vs. no funcionales

Aspecto	Requerimientos Funcionales	Requerimientos No Funcionales
Definición	Especifica lo que el sistema debe hacer (acciones y tareas).	Describe qué tan bien funciona el sistema (atributos de calidad).
Propósito	Garantizar que el sistema realice las funciones necesarias para los usuarios.	Asegurar que el sistema cumpla con los estándares de calidad para rendimiento, confiabilidad y usabilidad.
Enfoque	Se centra en las características y capacidades del sistema.	Se centra en la calidad del sistema y la experiencia del usuario.
Ejemplo	Inicio de sesión del usuario, recuperación de datos, procesamiento de pagos.	Velocidad del rendimiento, disponibilidad del sistema, protocolos de seguridad.
Método de Prueba	Pruebas funcionales (para validar el comportamiento de la función).	Pruebas de rendimiento, seguridad y usabilidad.
Medición	Medido a través de funciones o acciones específicas.	Medido por atributos de calidad (por ejemplo, tiempo de respuesta, tiempo de actividad).
Impacto en el Desarrollo	Definido temprano para dar forma a la funcionalidad central del sistema.	A menudo se refina durante el desarrollo para garantizar la estabilidad y escalabilidad del sistema.
Objetivo	Satisfacer las necesidades del usuario y del negocio en cuanto a capacidades del sistema.	Mejorar la experiencia del usuario y garantizar la solidez del sistema.
Importancia	Proporciona funciones esenciales para cumplir con los requisitos del usuario.	Garantiza que el sistema sea eficiente, confiable y fácil de usar.

Nota: Tomado de Visure (2025)

Ejemplo:

El sistema debe analizar ensayos estudiantiles para evaluar coherencia textual, con un tiempo de respuesta inferior a 2 segundos por texto.

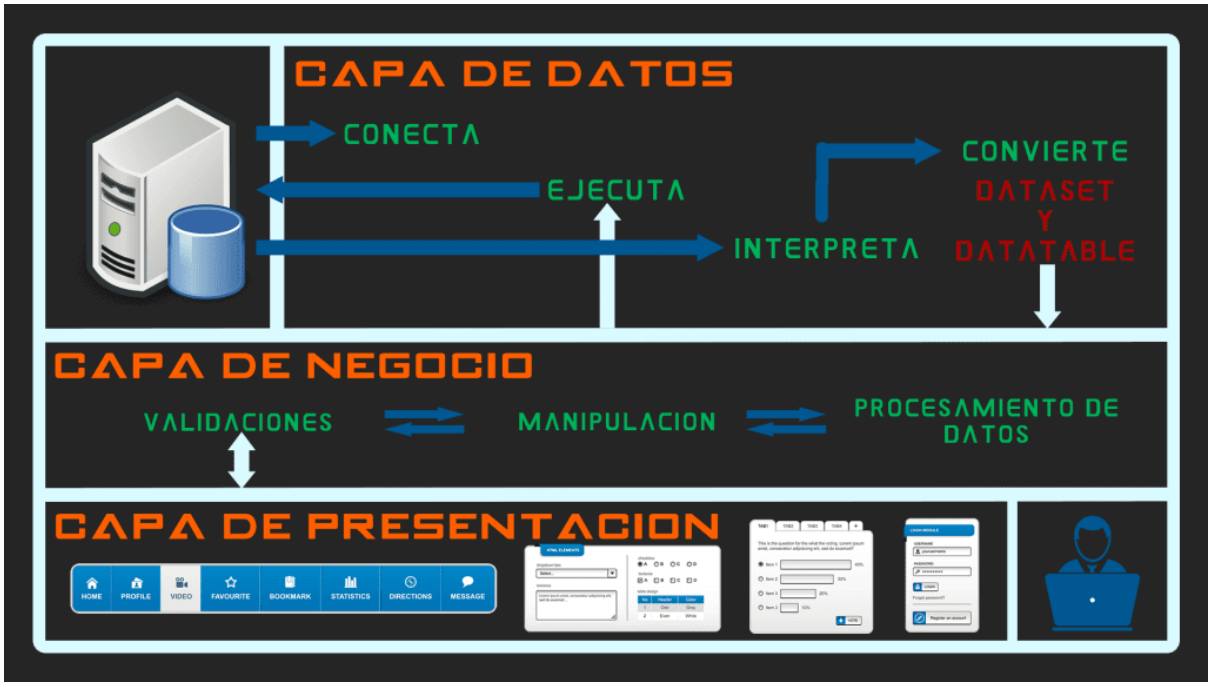
15.3.2. Diseño arquitectónico

Define la estructura general del sistema y cómo se integrarán sus componentes principales.

- Arquitectura en Capas (Figura 9)
    - Capa de presentación: Interfaz de usuario accesible e intuitiva.

- Capa lógica: Procesamiento del lenguaje, incluyendo análisis semántico y sintáctico.
- Capa de datos: Bases de datos con textos educativos, modelos entrenados y resultados de análisis.

Figura 9. Arquitectura en capas



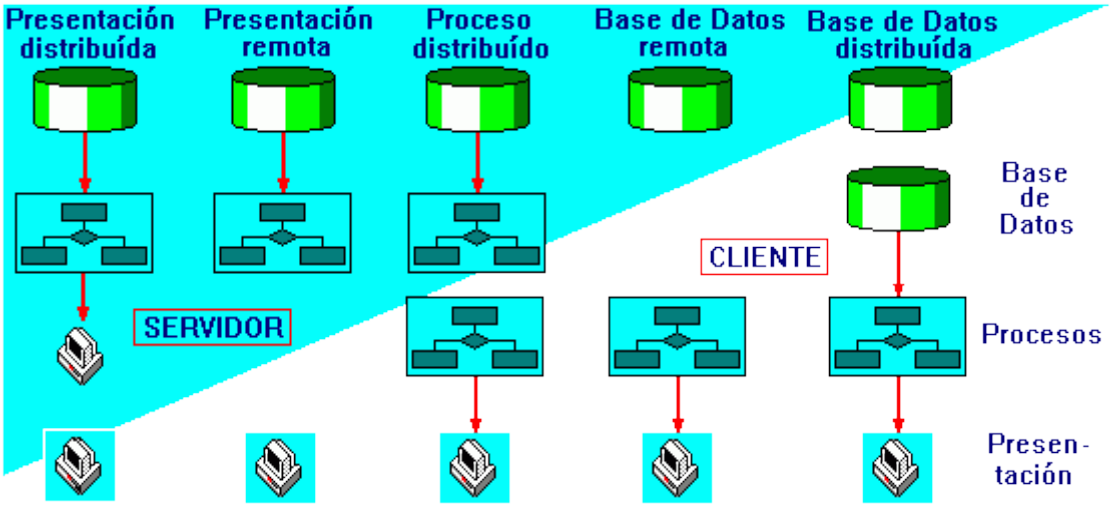
Nota: Tomado de videojuegosdesarrollo (2025)

Ejemplo:

Se adoptará una arquitectura en capas, donde la interfaz web se conecta a una API RESTful (Red Hat, 2025) *que accede al modelo de PLN alojado en la nube.*

- Arquitectura Distribuida (Figura 10)
  - Asegura que el sistema pueda manejar grandes volúmenes de datos de manera eficiente.

Figura 10. Arquitectura distribuida



Nota: Tomado de ittgweb (2016)

Ejemplo:

El proyecto utilizará una arquitectura distribuida en AWS, con instancias elásticas para el procesamiento de datos y almacenamiento en S3 (Amazon, 2025b).

15.3.3. Diseño de modelos de PLN

La selección y diseño del modelo de PLN es fundamental para el éxito del proyecto.

- Elección del Modelo
    - Modelos preentrenados como BERT, GPT o RoBERTa pueden adaptarse a necesidades educativas específicas.

Ejemplo:

Se empleará BERT multilingüe para la comprensión semántica de textos en español e inglés.

- Arquitectura del Modelo
    - Definir la cantidad de capas, tamaño del vocabulario y técnicas de fine-tuning.

Ejemplo:

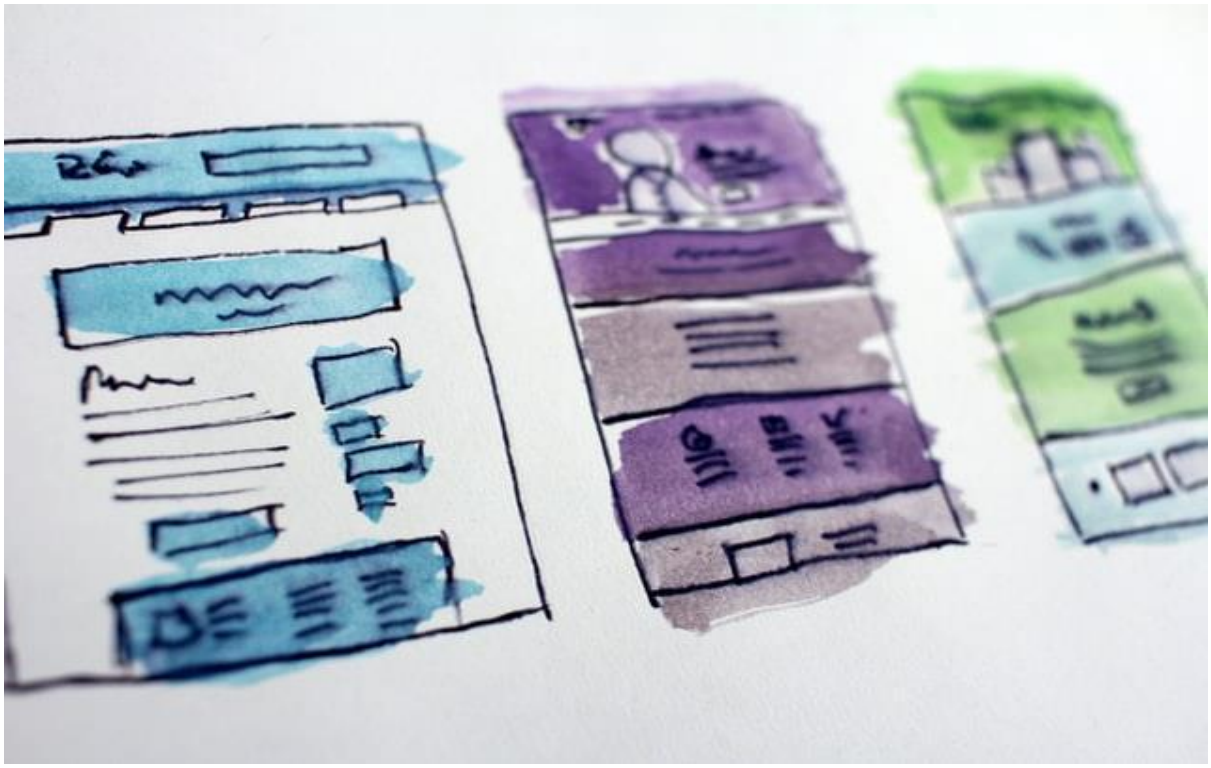
El modelo BERT se ajustará utilizando un corpus de textos educativos con técnicas de aprendizaje supervisado para tareas de clasificación textual.

15.3.4. Diseño de la interfaz de usuario (UI)

El diseño de la interfaz debe ser intuitivo y adaptado a las necesidades del usuario final (Figura 11).

- Principios de Diseño UI
    - Simplicidad, accesibilidad y consistencia visual.
- Prototipado
    - Creación de prototipos de baja y alta fidelidad para iterar sobre el diseño.

Figura 11. Diseño de interfaz de usuario



Nota: Tomado de Yeepli (2021)

Ejemplo:

Se diseñará una interfaz web que permitirá a los docentes cargar ensayos y recibir análisis automáticos en tiempo real.

15.3.5. Estrategia de integración de componentes

Se debe definir cómo los diferentes módulos del sistema interactuarán entre sí.

- Integración de API
    - Uso de APIs para la conexión entre la interfaz, el motor de PLN y las bases de datos.

Ejemplo:

La interfaz se integrará a través de una API que enviará los textos al motor de PLN y recuperará los resultados del análisis.

- Integración de Modelos en la Nube
    - Los modelos de PLN se desplegarán en plataformas en la nube para garantizar acceso y escalabilidad.

Ejemplo:

Los modelos se desplegarán en Google Cloud AI Platform, permitiendo el acceso desde múltiples dispositivos y entornos educativos.

15.3.6. Plan de validación del diseño

Un buen diseño incluye planes de validación para asegurar su efectividad antes de la implementación completa.

- Pruebas de Usuario: Validar la usabilidad y experiencia del usuario.
- Pruebas de Rendimiento: Verificar tiempos de respuesta y manejo de cargas de datos.
- Pruebas de Precisión del Modelo: Medir métricas clave como precisión, recall y F1-score.

Ejemplo:

Se realizarán pruebas piloto con 50 docentes para validar la precisión del modelo y la experiencia de usuario de la plataforma.

15.4 Herramientas

La selección adecuada de herramientas para un proyecto de Procesamiento de Lenguaje Natural (PLN) en el campo educativo es crucial para garantizar su éxito. Estas herramientas abarcan desde lenguajes de programación, bibliotecas especializadas y plataformas de desarrollo, hasta entornos de despliegue y visualización de resultados. Un proceso de selección eficaz debe considerar factores como los objetivos del proyecto, la complejidad de los datos, la escalabilidad y los recursos disponibles. Esta sección proporciona una guía técnica y profesional para elegir las herramientas más adecuadas, con ejemplos específicos para proyectos educativos.

15.4.1. Lenguajes de programación

El lenguaje de programación elegido debe ser robusto, tener soporte comunitario y ofrecer bibliotecas especializadas en PLN.

- Python: Es el lenguaje más utilizado en PLN debido a su sintaxis simple y su ecosistema de bibliotecas. Ofrece frameworks como NLTK, SpaCy, TensorFlow y PyTorch.
- R: Adecuado para análisis estadísticos y minería de texto, aunque menos potente que Python para tareas avanzadas de PLN.

Ejemplo:

Se seleccionará Python por su compatibilidad con bibliotecas de PLN y aprendizaje automático, utilizando Jupyter Notebooks (Jupyter, 2025) *para el desarrollo iterativo.*

15.4.2. Bibliotecas y frameworks de PLN

Las bibliotecas y frameworks proporcionan funcionalidades esenciales para el procesamiento, análisis y modelado del lenguaje (Figura 13).

- NLTK (Natural Language Toolkit): Ideal para proyectos educativos básicos que requieren procesamiento de texto, tokenización y análisis sintáctico.
- SpaCy: Recomendado para proyectos más avanzados por su alto rendimiento en procesamiento de texto.
- Hugging Face Transformers: Ofrece modelos preentrenados como BERT, GPT y RoBERTa, ideales para tareas complejas de PLN.

Figura 13. Biblioteca y frameworks

Nota: Tomado de Laprovittera (2025)

Ejemplo:

Se utilizará SpaCy para preprocesamiento eficiente de textos y Hugging Face para implementar modelos de comprensión lectora en español.

15.4.3. Plataformas de desarrollo y entrenamiento

La plataforma de desarrollo debe permitir un entorno colaborativo, escalable y con acceso a recursos computacionales.

- Google Colab (Google, 2025b): Proporciona entornos gratuitos con acceso a GPUs, ideal para prototipado y desarrollo rápido.
- Kaggle (Kaggle, 2025): Ofrece notebooks en la nube con acceso a datasets públicos y recursos de computación.
- AWS SageMaker (Amazon, 2025c): Adecuado para proyectos a gran escala, ofreciendo capacidades avanzadas de entrenamiento y despliegue.

Ejemplo:

El desarrollo inicial se realizará en Google Colab para aprovechar recursos gratuitos, migrando a AWS SageMaker en la fase de producción.

15.4.4. Herramientas de almacenamiento y gestión de datos

Los datos lingüísticos requieren almacenamiento seguro y accesible (Figura 14).

- MongoDB (MongoDB, 2025): Base de datos NoSQL eficiente para almacenar textos no estructurados.

- PostgreSQL (PostgreSQL, 2025): Ofrece robustez para gestionar datos estructurados y metadatos.
- Google Cloud Storage / AWS S3: Para almacenamiento en la nube escalable.

Figura 14. Herramientas de gestión de datos

Nota: Tomado de EDTeam (2020)

Ejemplo:

Los corpus de texto se almacenarán en MongoDB para un acceso rápido, mientras que los modelos entrenados se alojarán en AWS S3.

15.4.5. Plataformas de despliegue

El despliegue efectivo de modelos y plataformas educativas debe garantizar accesibilidad y rendimiento.

- Heroku (Salesforces, 2025): Ideal para el despliegue rápido de aplicaciones web educativas.
- Docker (Docker, 2025): Asegura la portabilidad y escalabilidad del sistema mediante contenedores.
- Google Cloud AI Platform: Proporciona entornos robustos para despliegue de modelos de aprendizaje automático.

Ejemplo:

La aplicación educativa se desplegará en Heroku para prototipado inicial, utilizando Docker para facilitar su portabilidad a entornos más robustos.

15.4.6. Herramientas de Evaluación y Visualización

Es esencial medir el rendimiento del sistema y presentar resultados de manera clara.

- Scikit-learn: Proporciona métricas de evaluación como precisión, recall y F1-score.
- Matplotlib (Matplotlib, 2024) y Seaborn (Waskom, 2024): Herramientas para visualización de datos.
- TensorBoard (TensorFlow, 2025b): Permite el monitoreo del rendimiento del modelo durante el entrenamiento.

Ejemplo:

Scikit-learn se utilizará para evaluar el rendimiento del modelo, mientras que TensorBoard ayudará a visualizar métricas durante el entrenamiento.

15.5. Cronograma con recursos

El cronograma con recursos es un componente esencial para la gestión efectiva de proyectos de Procesamiento de Lenguaje Natural (PLN) en educación. Este documento detalla las actividades del proyecto, sus tiempos de ejecución y los recursos necesarios para cada fase. Un cronograma bien estructurado asegura que las tareas se completen a tiempo, dentro del presupuesto y con la calidad esperada. Esta sección describe el proceso de elaboración de un cronograma con recursos, con información técnica y ejemplos específicos.

15.5.1. Identificación de fases del proyecto

Dividir el proyecto en fases facilita la planificación y asignación de recursos:

1. Planificación: Definición de objetivos, recopilación de requerimientos.
2. Recolección y Preprocesamiento de Datos: Obtención de corpus, limpieza y normalización.
3. Desarrollo del Modelo de PLN: Selección, entrenamiento y validación del modelo.
4. Despliegue: Implementación de la solución en entornos educativos.
5. Evaluación y Ajustes: Validación final y optimización.

Ejemplo:

La fase de desarrollo del modelo se estima en 8 semanas, con un científico de datos y un lingüista computacional dedicados a tiempo completo.

15.5.2. Estimación de duración y secuenciación de tareas

Cada tarea debe tener una duración estimada y estar ordenada de forma lógica.

Ejemplo de cronograma simple:

- Planificación: 2 semanas
- Recolección de datos: 3 semanas
- Preprocesamiento: 2 semanas
- Desarrollo del modelo: 8 semanas
- Evaluación y ajuste: 3 semanas

15.5.3. Asignación de recursos

- Recursos Humanos: Científicos de datos, especialistas en PLN, docentes.
- Tecnológicos: Plataformas como Google Colab, AWS, bibliotecas como SpaCy y TensorFlow.
- Financieros: Presupuesto para infraestructura, licencias y remuneraciones.

Ejemplo:

Durante la fase de despliegue, se necesitarán recursos en la nube (AWS S3) y un ingeniero DevOps a tiempo parcial.

15.5.4. Herramientas para la elaboración del cronograma

- Microsoft Project: Para gestión avanzada de cronogramas.
- Trello: Gestión ágil con tableros visuales.
- GanttProject: Creación de diagramas de Gantt detallados.

Ejemplo:

Se utilizará GanttProject para visualizar las dependencias entre tareas y Microsoft Project para el monitoreo de recursos.

Referencias

Amazon. (2025a). *Amazon Web Services*. https://aws.amazon.com/

Amazon. (2025b). *Amazon S3*. https://aws.amazon.com/s3/

Amazon. (2025c). *Amazon SageMaker*. https://aws.amazon.com/sagemaker

Amblard-Ladurantie, C. (2024*). Gestión de riesgos: una práctica esencial para el éxito*. https://www.mega.com/es/blog/que-es-la-gestion-de-riesgo

Brito, P. (2015). *Manual del Investigador: Metodología de la investigación: lo cuantitativo y cualitativo*. https://metodologiaecs.wordpress.com/2015/04/18/manual-del-investigador-metodologia-de-la-investigacion-lo-cuantitativo-y-cualitativo/

Docker. (2025). https://www.docker.com/

EDTeam. (2020). *Motores de bases de datos NoSQL*. https://ed.team/comunidad/motores-de-bases-de-datos-nosql

Explosion. (2025). *spaCy*. https://spacy.io/

Google. (2025a). *Google Cloud*. https://cloud.google.com/

Google. (2025b). *Google Colab*. https://colab.research.google.com/

Hugging Face. (2025). https://huggingface.co/

ittgweb. (2016). *Diseño de Software de Arquitectura Distribuida*. https://ittgweb.wordpress.com/2016/05/29/3-6-diseno-de-software-de-arquitectura-distribuida/

Jupyter. (2025). *Jupyter Notebook*. https://jupyter.org/

Kaggle. (2025). https://www.kaggle.com/

Kinsta. (2022*). ¿Qué Es el Web Scraping? Cómo Extraer Legalmente el Contenido de la Web*. https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/

Laprovittera, C. (2025). *Las 100 Mejores Bibliotecas de Python*. https://achirou.com/bibliotecas-de-python/

Matplotlib. (2024). *Matplotlib: Visualization with Python*. https://matplotlib.org/

MongoDB. (2025). https://www.mongodb.com/

NLTK Project. (2024). *Natural Language Toolkit*. https://www.nltk.org/

NumFOCUS. (2024). *pandas*. https://pandas.pydata.org/

PostgreSQL. (2025). https://www.postgresql.org/

Red Hat. (2025). *What is a REST API?* https://www.redhat.com/en/topics/api/what-is-a-rest-api

Rodríguez, V. (2024). *¿Qué son los cronogramas de actividades y cómo crear uno?* https://www.systec-consulting.net/post/qu%C3%A9-son-los-cronogramas-de-actividades-y-c%C3%B3mo-crear-uno

Salesforce. (2025). *Heroku*. https://www.heroku.com/

Scikit-learn. (2025). https://scikit-learn.org/stable/about.html

TensorFlow. (2025a). https://www.tensorflow.org/

TensorFlow. (2025b*). TensorBoard: TensorFlow's visualization toolkit*. https://www.tensorflow.org/tensorboard

The PyTorch Foundation. (2025). *PyTorch*. https://pytorch.org/

videojuegosdesarrollo. (2025). *Arquitectura en Capas*. https://www.videojuegosydesarrollo.com/arquitectura-de-desarrollo-en-capas/

Visure. (2025). *Requisitos funcionales y no funcionales*. https://visuresolutions.com/es/gu%C3%ADa-de-trazabilidad-de-gesti%C3%B3n-de-requisitos/requisitos-funcionales-vs-no-funcionales/

Waskom, M. (2024). *seaborn: statistical data visualization*. https://seaborn.pydata.org/

Weston, G. (2023). *TensorFlow vs PyTorch – Key Differences*. https://101blockchains.com/tensorflow-vs-pytorch/

Yeepli. (2021). *Las 8 nuevas tendencias en el diseño de interfaz de usuario*. https://yeeply.com/blog/diseno/interfaz-de-usuario-8-tendencias/

Términos para definición

Web scraping

El web scraping es una técnica utilizada para extraer información de sitios web de manera automatizada. Consiste en el uso de programas o scripts que navegan por páginas web, recopilan datos específicos y los almacenan en formatos estructurados, como hojas de cálculo o bases de datos. Esta técnica se basa en la simulación de la navegación humana en la web, pero con la capacidad de procesar grandes volúmenes de información en menos tiempo. El web scraping es fundamental en el análisis de datos, ya que permite acceder a información que no se encuentra disponible a través de APIs o bases de datos públicas. En el ámbito del Procesamiento de Lenguaje Natural (PLN), esta técnica se emplea para recopilar corpus lingüísticos, artículos, opiniones o cualquier contenido textual necesario para entrenar modelos. Los lenguajes de programación como Python ofrecen bibliotecas especializadas, como BeautifulSoup, Scrapy y Selenium, que facilitan la implementación del web scraping. Es importante destacar que, aunque es una herramienta poderosa, su uso debe realizarse respetando las normativas legales y éticas, como las políticas de uso de los sitios web y las regulaciones sobre propiedad intelectual y privacidad de los datos.

Escalabilidad

La escalabilidad es la capacidad que tiene un sistema, red o proceso para manejar un aumento en la carga de trabajo sin comprometer su rendimiento, o para ampliarse de manera eficiente cuando se incrementan los recursos. En el contexto del desarrollo de software y proyectos tecnológicos, como aquellos basados en Procesamiento de Lenguaje Natural (PLN), la escalabilidad es fundamental porque asegura que una solución pueda adaptarse al crecimiento en la cantidad de usuarios, datos o complejidad de las tareas, sin requerir rediseños significativos. Existen dos tipos principales de escalabilidad: vertical y horizontal. La escalabilidad vertical implica mejorar la capacidad de un único servidor o recurso, mientras que la horizontal se refiere a la adición de más unidades o nodos al sistema. Esta última es común en entornos de computación en la nube, donde se pueden añadir recursos bajo demanda. La escalabilidad también está relacionada con la eficiencia en costos, ya que un sistema escalable permite optimizar recursos sin incurrir en gastos innecesarios. En proyectos educativos basados en PLN, por ejemplo, la escalabilidad garantiza que plataformas de análisis de texto o tutoría inteligente puedan atender a más estudiantes o analizar volúmenes mayores de información sin degradar la calidad del servicio.

PUCE Virtual

EDUCACIÓN FLEXIBLE

La excelencia no se improvisa

síguenos