# Actas Capitulares de la Habana

Edian Broche Castro
Roger Fuentes Rodríguez
Kevin Manzano Rodríguez
Massiel Paz Otaño
Jackson Claudio Vera Pineda

## Índice



## 1. Introducción

### 1.1. Objetivo del Proyecto

Se tiene como objetivo desarrollar una propuesta de transcripción y procesamiento de un conjunto de imagenes, para su posterior uso como dataset en futuros proyectos para identificar entidades nombradas y crear grafos de conocimiento.

### 1.2. Contexto

Se tiene una serie documental perteneciente al fondo Ayuntamiento de La Habana del Archivo Histórico de la Oficina del Historiador de la Ciudad de la Habana.

Esta serie documental se divide en dos grupos o subseries: los libros originales (1550 - 1898) y los libros trasuntados (1550 - 1809). Los primeros destacan por su riqueza de contenido y forma; los segundos por dejar constancia de la labor del Ayuntamiento para garantizar la perdurabilidad en el tiempo de este tipo documental, al ser copias realizadas en la segunda mitad del siglo XIX.

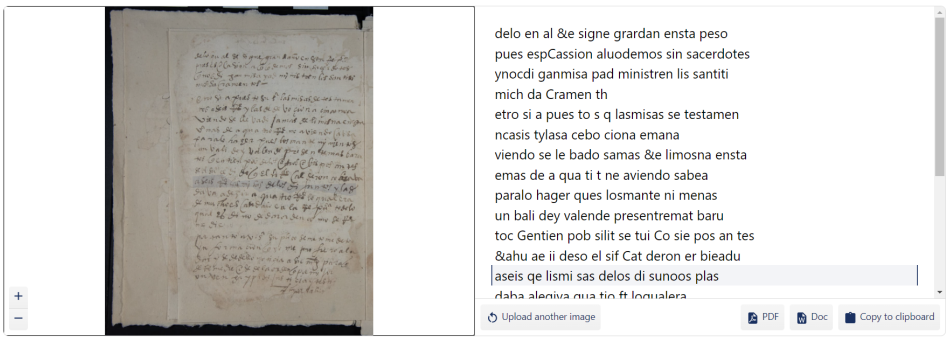
Las actas dejan la huella de una institución colonial y su devenir en el tiempo: el Ayuntamiento. Ellas recogen los planteamientos y discusiones de aquellos problemas que interesaron a los pobladores del lugar, ya fueran de índole económica, política o social; están reflejados los hechos mas significativos de cada una de las épocas.

Se tiene como corpus el tomo 1 digitalizado, el resto está en proceso de digitalización.

## 2. Metodología

### 2.1. Enfoque del Proyecto

Existen múltiples herramientas para la transcripción de documentos históricos entre estas destacan Transkribus la cual fue recomendada por el cliente por su uso en anteriores trabajos, pero debido al deterioro de los documentos además del tipo específico de tipografía los resultados no eran satisfactorios. Debido al bajo **accuracy** provisto por esta herramienta creada por especialistas en el tema podemos asumir que el problema es bastante complejo.




Nos encontramos con dos grandes problemas, falta de datos para llevar a cabo un entrenamiento de algún modelo pues muchos de los datasets encontrados no poseen el idioma requerido (Español) ni el tipo de letra (procesal-

cortesana), y una tarea que solo un ojo humano especializado podría realizar (tipógrafo)

Debido a todo lo expuesto el proyecto fue enfocado en una búsqueda exhaustiva de datos clasificados, un procesamiento minucioso de las imágenes, el uso de varios OCRs, y un posprocesamiento con diccionarios del idioma y LLM.

### 2.2. Recursos Utilizados

Como lenguaje principal se utilizo python, debido a que brinda un fácil acceso a librerias de procesamiento de texto e imágenes. Entre las librerias utilizadas destacan **Kraken**, **PIL**, **TensorFlow**, **Numpy**, **matplotlib**, **scipy**, **spacy**, **symspellpy** y **transformers**.

El LLM utilizado fue **Gemini** por cuestiones económicas, como diccionario de frequencias utilizamos spanish frequency dictionary (se considero la creación de un diccionario a partir de un libro escrito por Luis XV pero se tendrían muy pocas palabras en comparación con el utilizado), el modelo de spacy utilizado fue **es core news sm**.

Entre los datos recopilados inicialmente se utilizó dataset de rodrigo es un dataset de Español antiguo, pero la tipografía no coincide, en este caso se utiliza letra gótica, luego se encontró todo un corpus de documentos en Español de años anteriores a 1900 CODEA y con una estructura que podiamos aprovechar, la mayoría de los documentos presentan una triple representación, la imagen, una transcripción paleográfica, y una presentanción crítica. El dataset inicialmente tiene alrededor de 4000 documentos, pero luego de filtrar por el tipo de letra requerida se reduce a 546, recalcando la dificultad de los datos.

AQUI FALTAN AGREGAR COSAS

## 3. Resultados

### 3.1. Logros Principales

En la primera etapa del producto se logra una mejora considerable en la imagen, se utiliza un pipeline para limpiar las imágenes con técnicas de filtrado como filtro Gaussiano para reducir el ruido general de la imagen y un filtro de mediana para reducir el ruido sal y pimienta. Se emplea un algoritmo de detección de bordes Canny para identificar letras presentes en los documentos, pero esto solamente deja los bordes por lo que se aplican luego operaciones morfológicas de dilatación y erosión para reconstruir el trazo. Se intento el uso de algoritmos como sobel y laplaciano , el último no funciono bien porque depende de la segunda derivada de los cambios pixel a pixel, y esto en documentos de texto donde el documento es blanco y las letras son negras y los cambios de papel tinta son muy abruptos indefine la derivada

y causo malos resultados. Además se probo binarización de la imagen, definiendo un umbral de intensidad donde si $x >$ threshold se toma como letra y en caso contrario como fondo , pero en zonas de sombra , probando con diferentes valores para threshold no se detecta bien la escritura, también se intento binarización adaptativa pero se recuperaba mucho ruido y manchas en el documento. Para el tratado de las manchas debido a la longevidad de los documentos(son amarillos) se implementó un conversor personalizado de la imagen de color a escala de grises ponderando las componentes rgb de forma diferente a lo usual (que es el promedio), r + g es el equivalente al amarillo por lo que se pondero r y g en menor grado jugando con los parámetros para tratar de contrarestar el ruido.

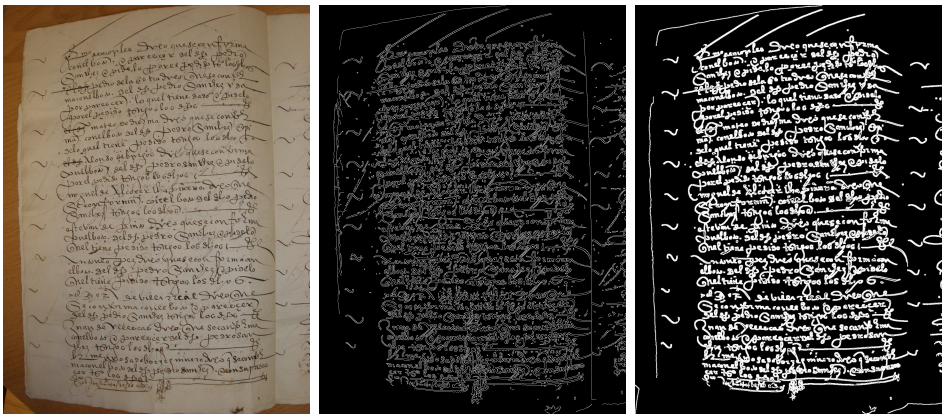


Figura 1: Original, detección de bordes, operaciones morfológicas de izquieda a derecha

Luego de binarizadas las imágenes son pasadas por dos procesos de segmentación, uno que brinda **Kraken** y una proyección de histogramas, se sugiere utilizar la primera pues otorga mejores resultados, y en caso de un proceso manual sería mejor utilizar **SAM2** pero no logramos automatizar el proceso, luego de segmentada la imagen esta es pasada a 3 OCRs distintos, **SimpleHTR**, **sinai-sam-rec-v4-best**, **McCATMuS-nfd-nofix-V1**, los cuales son modelos especializados en **handrwitten recognition**, los dos últimos provienen de Kraken, aquí tuvimos que tratarlos diferentes pues, el primero necesita la imagen segmentadas y los otros solamente los **bounding box**.

En la última etapa una vez procesadas la imágenes con las técnicas para mejorarlas y luego extraido el texto con los diferentes OCRs, se realizó un posprocesamiento para mejorar la calidad de las transcripciones, inspirados en el DataSet de CODEA, brindamos una transcripción crítica del documento. Dada la naturaleza de los documentos y los errores inherentes al OCR, implementamos un pipeline para corregir errores ortográficos y ajustar la chorencia del texto, de esta forma refinando la salida.

Se utilizó un modelo de lenguaje **Spacy** para segmentar el texto en tokens, para así identificar palabras y elementos no alfabéticos. Cada token fue corregido utilizando SymSpell, una herramienta basada en un diccionario de frecuencias, para separar correctamente palabras unidas y capitalizarlas. Tras las correcciones se utilizó un modelo generativo, Gemini para realizar un refinamiento semántico y estilístico; corrigiendo así el formato del texto y la gramática, además de mantener el contexto histórico del español antiguio. También se consideraron las diferentes salidas y se combinaron en una sola.

A continuación un ejemplo del flujo:

1. Essta es una prueva de ectraccion de teexto. Connoscida cosa sea a todos los queesta carta uieren como yo don Fferrando por la gracia de dios hey de Castiella

2. Esta es una prueba de extracción de texto. Conocida cosa sea a todos los que esta carta vieren como yo don Fferrando por la gracia de Dios hay de Castilla

3. Esta es una prueba de extracción de texto. Conocida cosa sea a todos aquellos que esta carta vieran, como yo, don Fernando, por la gracia de Dios, rey de Castilla.

### 3.2. Desafíos Superados

En el momento de la segmentación jugó un papel fundamental la búsqueda de parámetros adecuados para que tuviera mejor rendimiento los procesos posteriores.

Entre los desafíos superados en el posprocesamiento nos encontramos con la conservación del español antiguo pues no encontramos un diccionario de frecuencias de esa época, pero solventamos este problema con el LLM utilizado, la obtención del mismo fue compleja, pues se probaron con otros, por ejemplo **GPT-2**, **"google/mt5-small"**, **flax-community/spanish-t5-small**, pero los resultados fueron pésimos en los tres casos, además de que no teníamos un API cómoda y los modelos eran bastantes pesados, por lo que optamos por **Gemini**.

# 4. Conclusiones

## 4.1. Impacto del Proyecto

## 4.2. Recomendaciones para Futuras Implementaciones

# A. Anexos

## A.1. Documentación Adicional

## A.2. Códigos Fuente