

Proyecto EDAII: Aplicación de Algoritmos de Pattern Matching (KMP y Boyer-Moore)  
Sistema Inteligente de Detección Temprana de Reclamos Críticos en Textos de Atención al Cliente

Antecedentes:

En empresas de servicios (banca, telecomunicaciones, retail, transporte), los clientes envían diariamente grandes volúmenes de mensajes: correos, chats, tickets de soporte y formularios de reclamos.

Detectar patrones textuales críticos (amenazas legales, cancelaciones, estafas, reclamos graves) permite priorizar atención, reducir riesgos reputacionales y mejorar la experiencia del cliente.

El sistema se enfoca en búsqueda eficiente de patrones textuales, usando algoritmos clásicos de Pattern Matching, sin recurrir a IA avanzada.

Objetivo:

Diseñar un prototipo low-fidelity (mockups) de un sistema que simule la detección automática de reclamos críticos en textos de clientes, utilizando algoritmos KMP y Boyer-Moore, con énfasis en eficiencia, claridad algorítmica y análisis crítico.

Alcance en cuanto a la aplicación de los algoritmos:

KMP y Boyer-Moore para identificación eficiente de:

- Frases de reclamo
- Amenazas legales
- Intenciones de cancelación
- Insatisfacción severa

Opcional: Priorización de alertas según severidad

Importante: **no se evalúa IA**, sino uso correcto, eficiente y justificado de algoritmos clásicos.

**Fase 1:** Modelado del problema y diseño de entradas

- Revisión de conceptos: ¿Qué reclamos afectan la reputación de una empresa? ¿Son comunes las amenazas legales de un cliente? ¿Qué se suele hacer ante reclamos graves e insatisfacción severa (bajo qué contextos comúnmente se da esto)?
- Creación de una lista configurable de patrones a detectar: reclamos, amenazas, cancelaciones, etc.; por ejemplo:
  - “cancelaré el servicio”
  - “voy a demandar”
  - “esto es una estafa”
  - “fraude”
  - “reclamo formal”
  - “péssimo servicio”
  - “nunca solucionan nada”
- Ejemplos de entradas: mensajes de parte de los clientes.
- Clasificación de acuerdo a patrones detectados:
  - Queja leve
  - Reclamo
  - Reclamo crítico
  - Riesgo legal

- Primeros mockups: interfaz donde se cargan mensajes, lista de patrones, y resultados.
- Avance: Definición del problema, lógica/Diseño del sistema, conjunto de patrones, primer borrador de mockups.

Actividad principal de la fase 1:

1. Investigar (por internet o entrevistas breves) ejemplos de mensajes de reclamo en contextos empresariales.
2. Construir una tabla o archivo .txt con al menos 15 a 20 patrones separados por tipo o nivel de severidad para la posterior clasificación.
3. Clasificarlos si es posible (queja, reclamo, reclamo crítico, riesgo legal, etc.).
4. Simular mensajes reales (mensajes largos de clientes ficticios) donde algunos patrones estén ocultos -> existen datasets públicos de Amazon, por ejemplo, con este tipo de data?
5. Preparar esta lista de estos patrones para aplicar los algoritmos KMP y Boyer-Moore en la Fase 2. Puede ser una tabla en Excel con las columnas: Patrón, Categoría, Nivel de Alerta; o un archivo .csv con los patrones separados por salto de línea y columnas separadas por coma.

**Fase 2:** Diseño lógico del sistema y algoritmos

- Implementar en Java o Python el uso de KMP y/o Boyer-Moore sobre los mensajes, posibilidad de leer archivos con mensajes y patrones.
- Mostrar con ejemplos cómo se detecta un patrón en los mensajes y qué tan rápido funcionan los algoritmos comparándolos. ¿Qué algoritmo detecta más rápido? ¿Dónde falla? Qué hacer si hay más de un patrón encontrado?
- Establecer el flujo: entrada → detección → alerta → sugerencia de acción.
- Segundo borrador de mockups más refinado.

Avance: Algoritmo documentado, tabla de pruebas con ejemplos (salida de la ejecución de los algoritmos implementados), mockups completos.

Actividad principal de la fase 2:

Los estudiantes deberían probar sus algoritmos de búsqueda (KMP o Boyer-Moore) con textos reales o simulados, verificando si los patrones definidos en la Fase 1 son detectados correctamente (pruebas de manejo de mayúsculas vs minúsculas, tildes y signos de puntuación). Se debería tener al menos 15 pruebas, variando: casos positivos y negativos, patrón al inicio, en el medio o final, casos similares pero con errores gramaticales, mayúsculas/minúsculas, tildes, más de un patrón encontrado en el texto.

A continuación, se presenta una tabla de pruebas clara y explicativa, acompañada de ejemplos.

ID	Mensaje de Entrada (texto del cliente)	Patrón Buscado	Categoría	¿Detectado?	Posición(es)	Algoritmo Usado	Observación técnica
1	“Pésimo servicio, nunca solucionan nada.”	“péximo servicio”	Reclamo crítico	Sí	0	KMP	Patrón detectado al inicio del texto.
2	“Si esto no se arregla hoy, voy a demandar.”	“voy a demandar”	Riesgo legal	Sí	27	Boyer-Moore	Detección rápida por coincidencia desde el final.
3	“He llamado varias veces y nadie responde.”	“nadie responde”	Reclamo	Sí	28	KMP	Coincidencia exacta en mitad del texto.
4	“Cancelaré el servicio si no me dan solución.”	“cancelaré el servicio”	Reclamo crítico	Sí	0	Boyer-Moore	Patrón largo, BM realiza menos comparaciones.

<b>5</b>	“Esto parece una estafa, no confío en ustedes.”	“estafa”	Riesgo legal	Sí	18	KMP	Coincidencia parcial dentro de una frase más larga.
<b>6</b>	“El servicio no es bueno, es excelente.”	“no es bueno”	Queja leve	Sí	13	Boyer-Moore	Falso positivo semántico (limitación del algoritmo).
<b>7</b>	“PESIMO servicio, nadie contesta.”	“péssimo servicio”	Reclamo crítico	No	—	KMP	No detectado por diferencia en mayúsculas y tilde.
<b>8</b>	“Nunca solucionan nada, jamás vuelvo.”	“nunca solucionan nada”	Reclamo crítico	Sí	0	Boyer-Moore	Coincidencia exacta completa.
<b>9</b>	“Estoy pensando en cancelar el servicio.”	“cancelaré el servicio”	Reclamo crítico	No	—	KMP	Variante gramatical no detectada.
<b>10</b>	“Voy a demandar si no hay respuesta.”	“voy a demandar”	Riesgo legal	Sí	0	KMP	Patrón exacto al inicio.
<b>11</b>	“Fraude total, esto no es lo que ofrecieron.”	“fraude”	Riesgo legal	Sí	0	Boyer-Moore	Coincidencia temprana; BM eficiente.
<b>12</b>	“Este reclamo es formal y será escalado.”	“reclamo formal”	Reclamo	No	—	KMP	Orden distinto de palabras.
<b>13</b>	“Nunca, nunca solucionan nada.”	“nunca solucionan nada”	Reclamo crítico	Sí	7	Boyer-Moore	Patrón aparece después de repetición.
<b>14</b>	“Esto ya es una estafa y voy a demandar.”	“estafa”	Riesgo legal	Sí	13	KMP	Texto con múltiples patrones posibles.
<b>15</b>	“Esto ya es una estafa y voy a demandar.”	“voy a demandar”	Riesgo legal	Sí	27	Boyer-Moore	Más de un patrón detectado en el mismo texto.
<b>16</b>	“Gracias por la atención, todo quedó resuelto.”		Sin reclamo	No	—	KMP	Caso negativo correcto.

### Fase 3: Integración, documentación y presentación

El objetivo final para esta fase es finalizar el prototipo low-fidelity, integrar todos los componentes del sistema, analizar las limitaciones de los algoritmos KMP y Boyer-Moore en entornos reales, y presentar una visión crítica (consultar) sobre cómo estos podrían complementarse o superarse con técnicas más avanzadas (ej. inteligencia artificial, aprendizaje automático, análisis semántico).

#### 1. Integración del Sistema Simulado (Mockups)

Afinar la experiencia de usuario del sistema.

Simular cómo se cargarán mensajes, cómo se escanean con los algoritmos, y cómo se muestran las alertas.

Incluir por ejemplo, pantallas de:

  Ingreso de mensajes o emails.

  Configuración del conjunto de patrones.

  Panel de resultados con alertas (nivel de reclamo y sugerencia de acción).

  Panel administrativo (sugerencias, acciones, recomendaciones).

(Opcional) Pantalla de estadísticas del sistema (reclamos mayormente recibidos, frecuencia de mensajes de acuerdo a nivel de reclamo).

#### 2. Reflexión Crítica: Limitaciones de KMP/Boyer-Moore

En un apartado del informe con las ideas integradas en los mockups:

- a) Plantear preguntas como:

- ¿Qué no pueden detectar KMP y Boyer-Moore?
- ¿Qué ocurre si el mensaje tiene sinónimos o errores ortográficos?
- ¿Cómo manejarían patrones sarcásticos, ambiguos o irónicos?
- ¿Podrían confundir contexto positivo con negativo? (ej. "Este producto no es bueno, es excelente")
- ¿Qué mejoras podrían incorporar para que el sistema sea más inteligente?

b) Proponer ideas como:

- Uso de NLP, modelos de lenguaje, IA conversacional.
- Filtrado colaborativo (feedback de usuarios).
- Detección de sentimiento.
- Análisis semántico contextual.

c) “Versión con IA”:

Simulación completa del sistema: mockups navegables o enlazados: dónde está el componente inteligente?

### 3. Informe Final y Video

Documento final tipo informe ejecutivo:

- Nombre de su ‘empresa de desarrollo de sistemas computacionales’, misión, visión, logo...
- Introducción al problema, motivación, enfoque técnico.
- Conjunto de patrones.
- Pruebas de los algoritmos.
- Resultados y simulación del sistema.
- Mockups navegables.
- Análisis crítico de limitaciones y mejora con IA.
- Impacto ético e impacto empresarial.

Preparar video de presentación de 10-15 minutos con enfoque en el problema y solución propuesta y **compartirlo en el grupo de whatsapp**:

- Explicación clara del problema.
- Presentación del sistema.
- Demostración o recorrido de los mockups.
- Reflexión final + componente inteligente del sistema.

**NOTA: Subir en el aula virtual un archivo .zip: Mockups, informe y link al video-02/02/2026**

### Trabajo Opcional (Evaluación tipo Examen Final)

#### Implementación Completa y Funcional del Sistema de Detección de Reclamos

Este trabajo es voluntario y está dirigido a los grupos que deseen implementar completamente el sistema diseñado en el proyecto EDA II, pasando del prototipo conceptual (mockups) a un sistema de software funcional.

- La nota obtenida en la evaluación del sistema reemplaza o equivale a la nota del examen final.
- Solo se evaluarán implementaciones funcionales y correctas.
- No es obligatorio usar frameworks complejos ni bases de datos.

## **Objetivo del trabajo opcional**

Desarrollar una aplicación funcional que implemente correctamente los algoritmos KMP y Boyer–Moore para la detección de reclamos críticos en textos de atención al cliente, demostrando:

- Dominio algorítmico
- Buen diseño de software
- Correcta lectura y procesamiento de archivos (read, write)
- Capacidad de medición y comparación de desempeño
- Robustez en el manejo de entradas

## **Requisitos funcionales mínimos (obligatorios para quienes presenten el SW funcional)**

### **1 Entrada de datos**

El sistema debe permitir:

- Cargar mensajes desde:
  - Archivos .txt, Excel o .csv
- Cargar patrones desde:
  - Archivo externo (.txt, Excel o .csv)
- Manejar múltiples mensajes y múltiples patrones

OJO: *No se aceptan patrones quemados únicamente en el código.*

### **2 Gestión de patrones**

El sistema debe permitir editar dinámicamente los patrones:

- Agregar nuevos patrones
- Eliminar patrones existentes
- Clasificar patrones por:
  - Queja
  - Reclamo
  - Reclamo crítico
  - Riesgo legal

Los cambios deben reflejarse sin recomilar el programa (archivo externo o menú interactivo).

### **3 Implementación algorítmica**

El sistema debe implementar:

- KMP completo
- Boyer–Moore completo

Requisitos:

- Código propio (no librerías de búsqueda que ya implementen estos algoritmos, sí se pueden usar librerías/funciones para preprocessamiento o limpieza de texto como unidecode o que remueva puntuación)

## **4. Medición de rendimiento**

El sistema debe medir y mostrar:

- Tiempo de ejecución de:
  - KMP
  - Boyer–Moore
- Para el mismo mensaje y patrón

La medición puede ser:

- En milisegundos o nanosegundos
- Promedio de varias ejecuciones (opcional)

## **5 Interfaz del sistema (flexible)**

Se acepta cualquiera de las siguientes opciones:

- Interfaz gráfica simple (Swing, JavaFX, Tkinter, etc.)
- Interfaz web básica (opcional, no requerida)

Lo evaluado será:

- Claridad
- Usabilidad básica
- Flujo lógico

## **6 Manejo de errores (OBLIGATORIO)**

El sistema debe manejar correctamente:

- Archivos inexistentes
- Archivos vacíos
- Patrones no encontrados
- Mensajes sin reclamos

## **7 Entregables del trabajo opcional**

El grupo debe entregar un .zip o .rar con:

1. **Código fuente completo**
2. **Archivo(s) de patrones**
3. **Archivo(s) de mensajes**
4. **Documento técnico corto (5–7 páginas)** que incluya:
  - Nombre de su ‘empresa de desarrollo de sistemas computacionales’, misión, visión, logo...
  - Introducción al problema, motivación, enfoque técnico.

- Conjunto de patrones.
- Pruebas de los algoritmos.
- Arquitectura del sistema
- Descripción del flujo
- Capturas o evidencias de ejecución
- Análisis crítico de limitaciones y mejora con IA.
- Impacto ético e impacto empresarial.

**5. Demostración del sistema (presencial con la profesora)**

**6. Video corto (10–15 minutos compartido en el grupo de whatsapp):**

- Explicación clara del problema.
- Presentación del sistema y su funcionalidad.
- Reflexión final + componente inteligente del sistema.